Algorithms

Basic Coding

Winter 2022: Dan Calderone

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

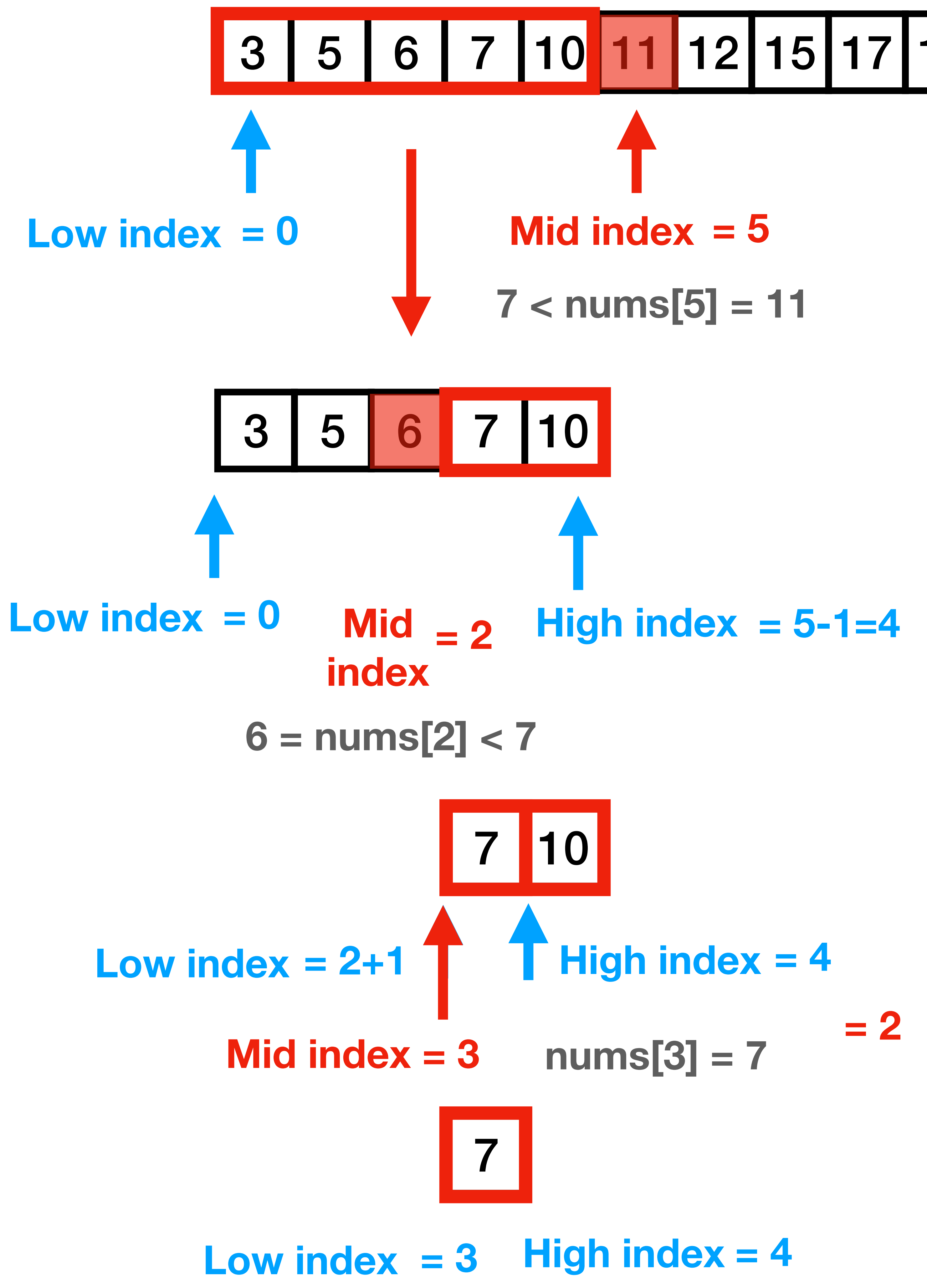
Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Binary Search Algorithm: (Divide and Conquer)

nums = [3, 5, 6, 7, 10, 11, 12, 15, 17, 19, 20, 21]; target = 7;

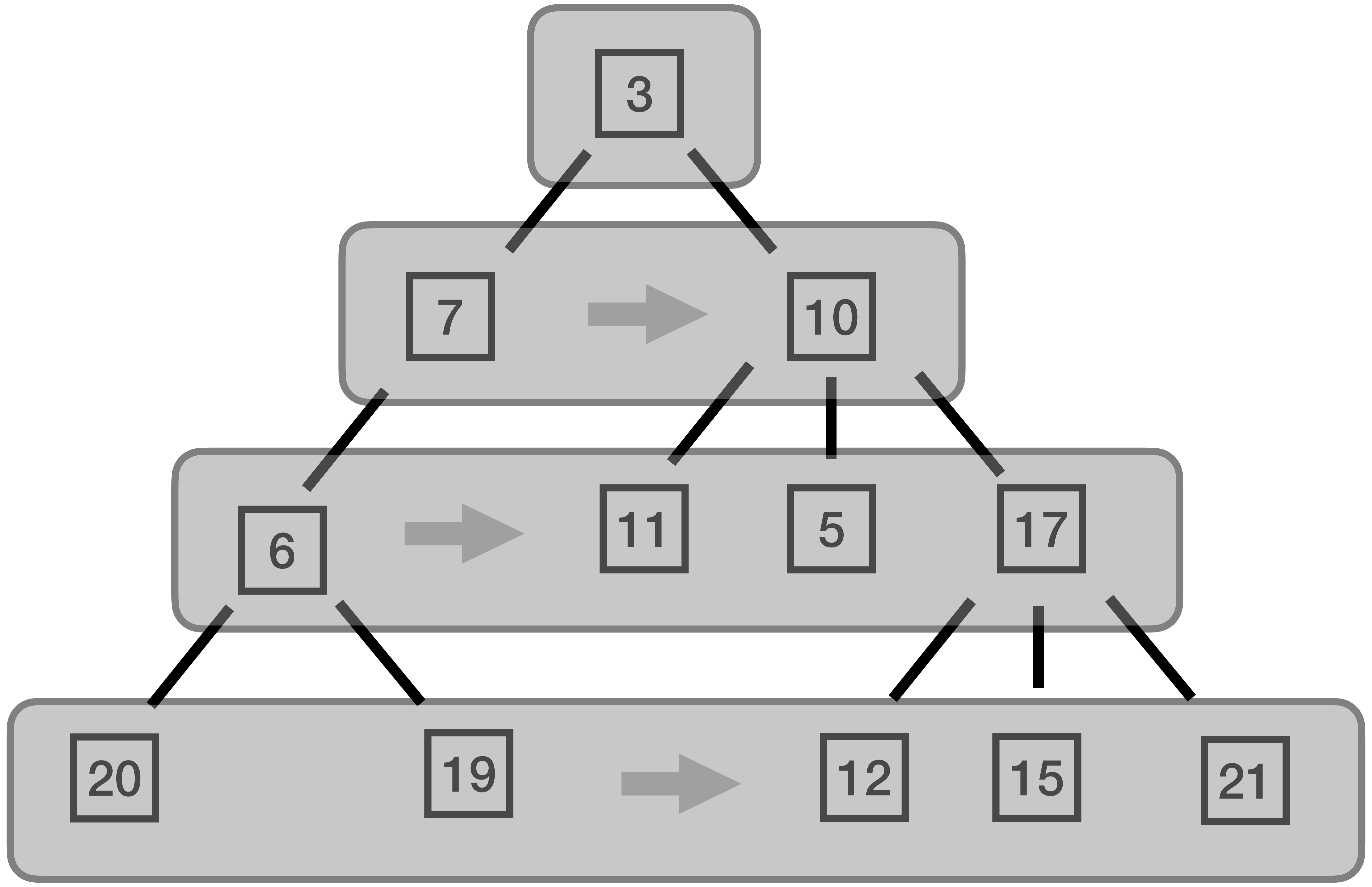


nums[3] = 7

... index = 3

Algorithms

Breadth First Search (BFS)



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

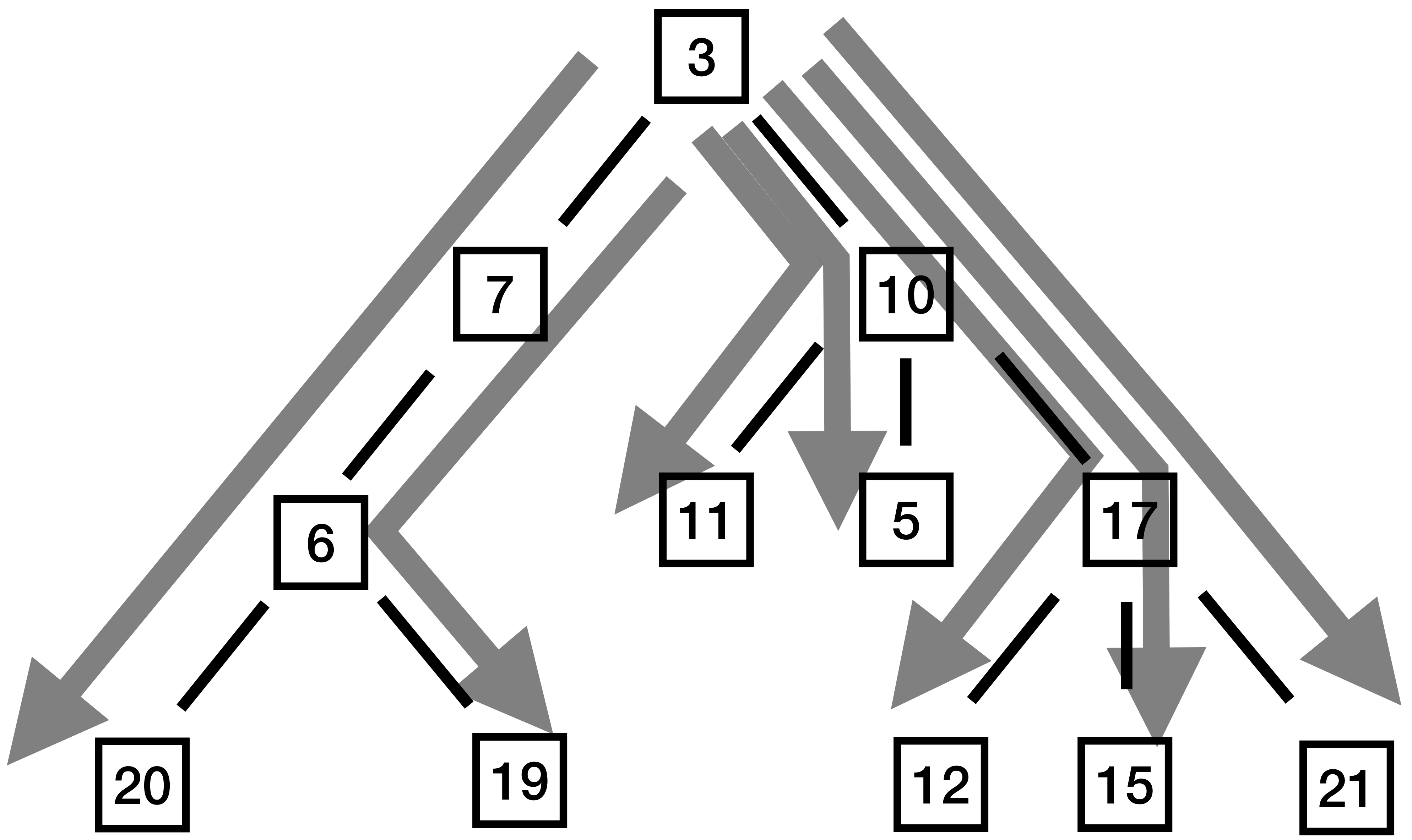
Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Depth First Search (DFS)



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Kadane's Algorithm

"find the maximum sum contiguous subarray"

- empty = 0
- all positive: solution is full array
- all negative: solution is empty array

Find the largest sum:

Initialize

best_sum = 0
current_sum = 0;

for each element x (left to right):
    current_sum = max(0,current_sum+x)
    best_sum = max(best_sum, current_sum)

Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

Fill Algorithm

- Flood Fill Algorithm

Kadane's Algorithm

"find the maximum sum contiguous subarray"

- empty = 0
- all positive: solution is full array
- all negative: solution is empty array

Find the largest sum w/ position

```
Initialize
best_sum = 0
current_sum = 0;
best_start = best_end = 0;
current_start = current_end = 0;

for each element x (left to right):
    If current_sum <= 0:
        current_start = current_end
        current_sum = x
    else:
        current_sum += x

    current_end++

    If current_sum > best_sum:
        best_sum = current_sum
        best_start = current_start
        best_end = current_end
```

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

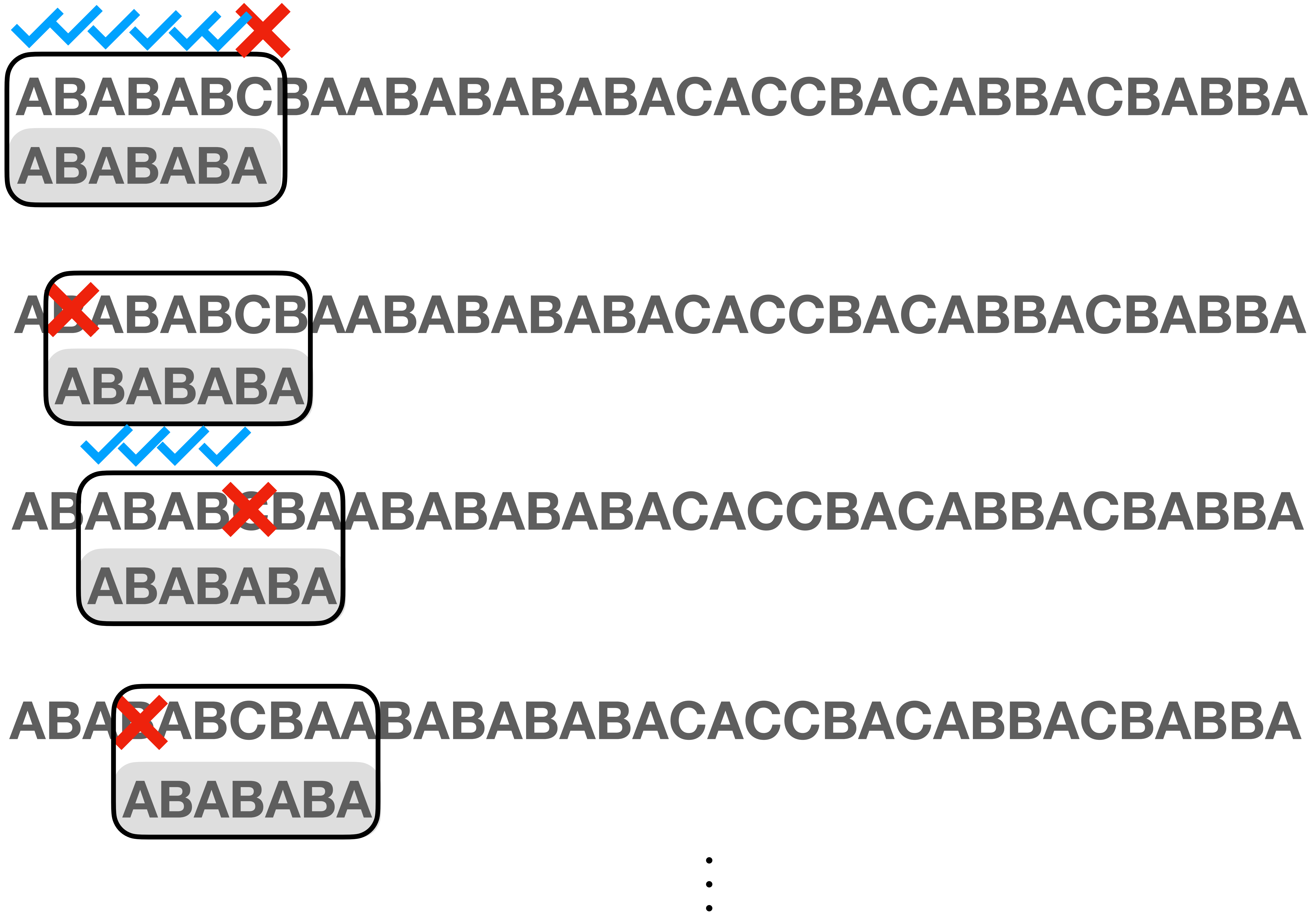
Fill Algorithm
- Flood Fill Algorithm

Naive Pattern Matching

Computation Cost:
Worst Case: **O(nm)**

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **AABABCBAABABABABACACCBACABBACBABBA**



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **AABABCBAABABABABACACCBACABBACBABBA**

preprocess the pattern to save time…

A B A B A B A

…prefix …suffix

Partial Match Table:

A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5

..first 1 char (A), *..first 2 char* (AB), *..first 6 char* (ABABAB), *..first 7 char* (ABABABA)

… **ABAB**

Proper prefixes	Proper Suffixes
A	B
AB	AB
ABA	BAB

ABABA

Proper prefixes	Proper Suffixes
A	A
AB	BA
ABA	ABA
ABAB	BABA

longest proper prefix that is also a suffix"

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

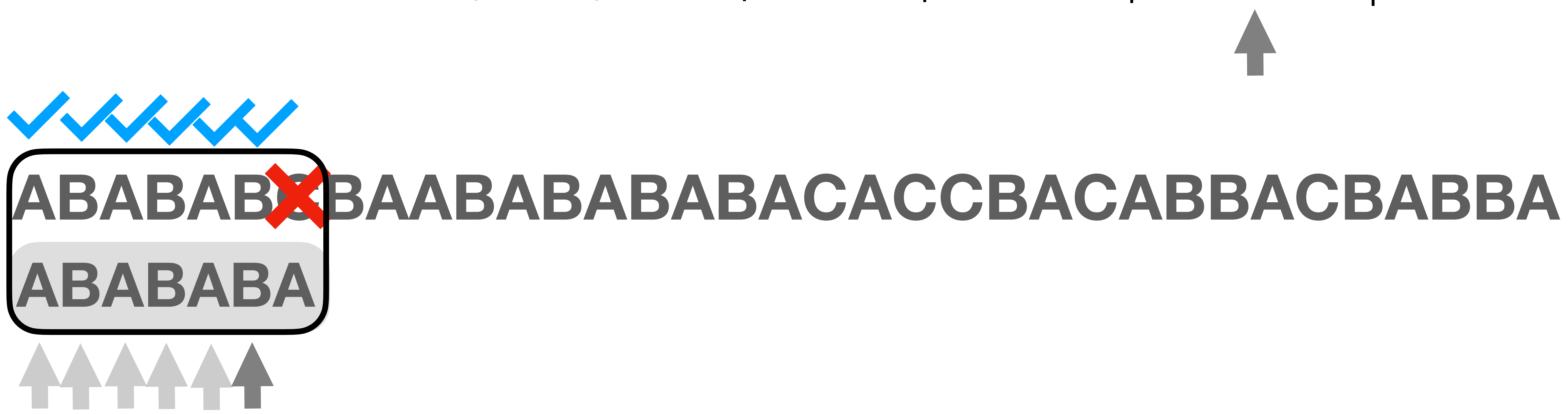
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

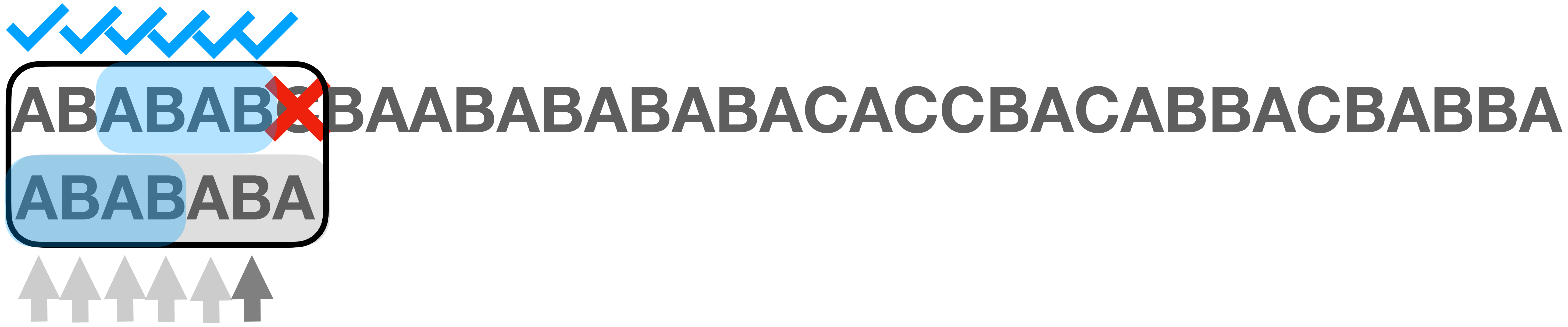
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

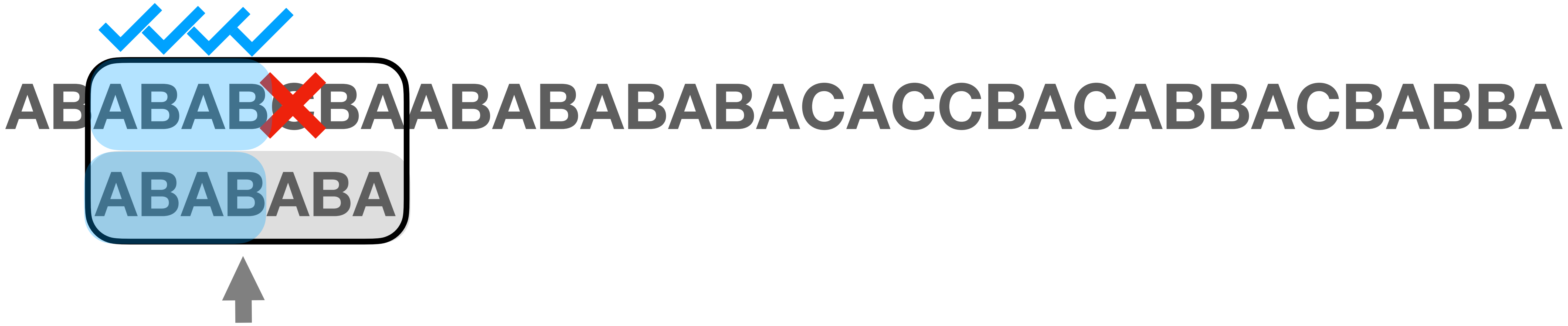
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

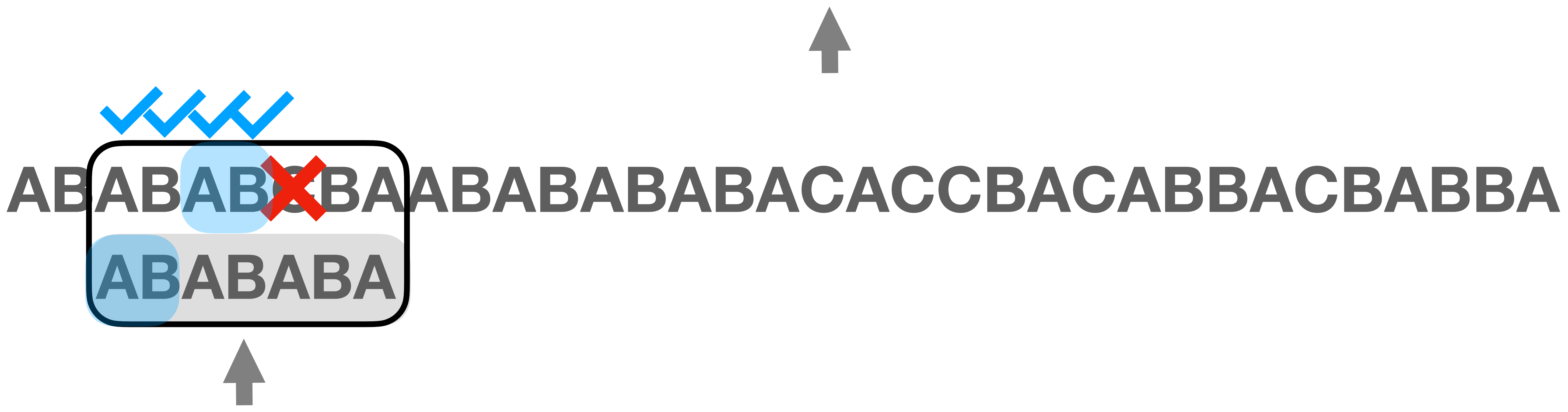
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A (..first 1 char)	AB (..first 2 char)	ABA	ABAB	ABABA	ABABAB (..first 6 char)	ABABABA (..first 7 char)
0	0	1	2	3	4	5

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5

Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

Fill Algorithm

- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

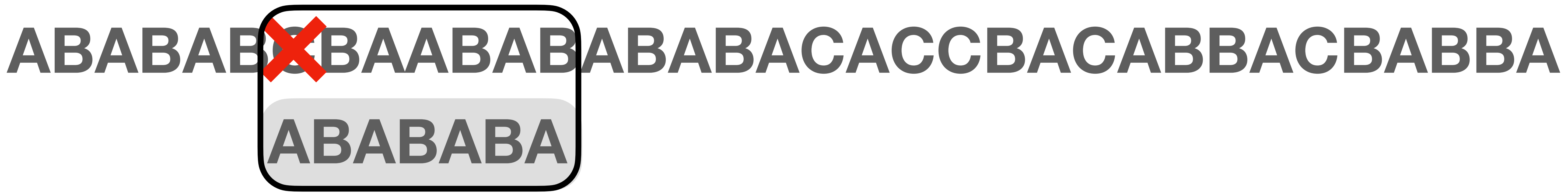
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

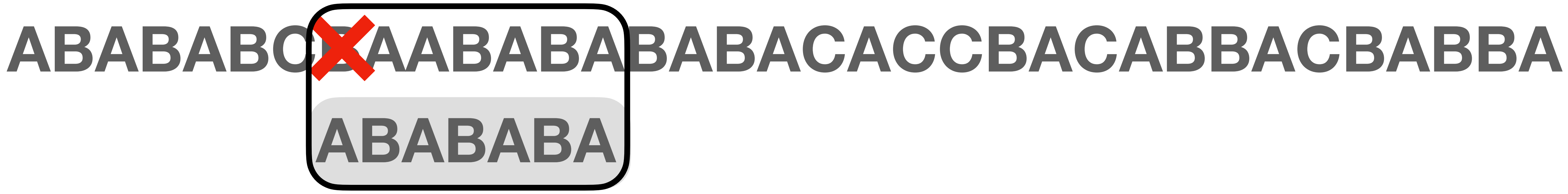
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A *..first 1 char*	AB *..first 2 char*	ABA	ABAB	ABABA	ABABAB *..first 6 char*	ABABABA *..first 7 char*
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

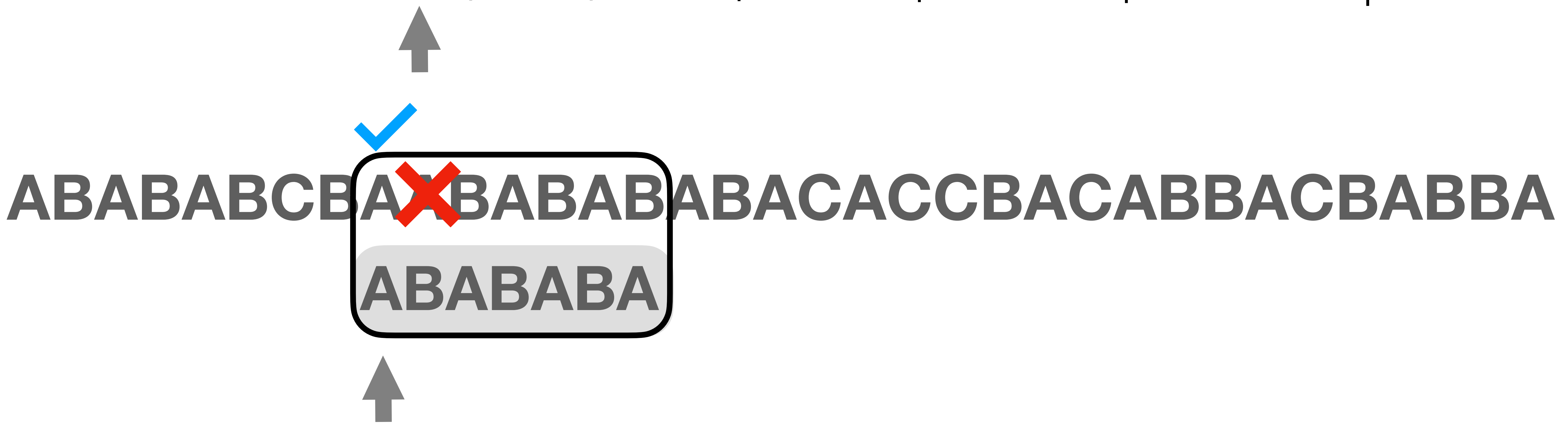
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A (..first 1 char)	AB (..first 2 char)	ABA	ABAB	ABABA	ABABAB (..first 6 char)	ABABABA (..first 7 char)
0	0	1	2	3	4	5



Algorithms

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

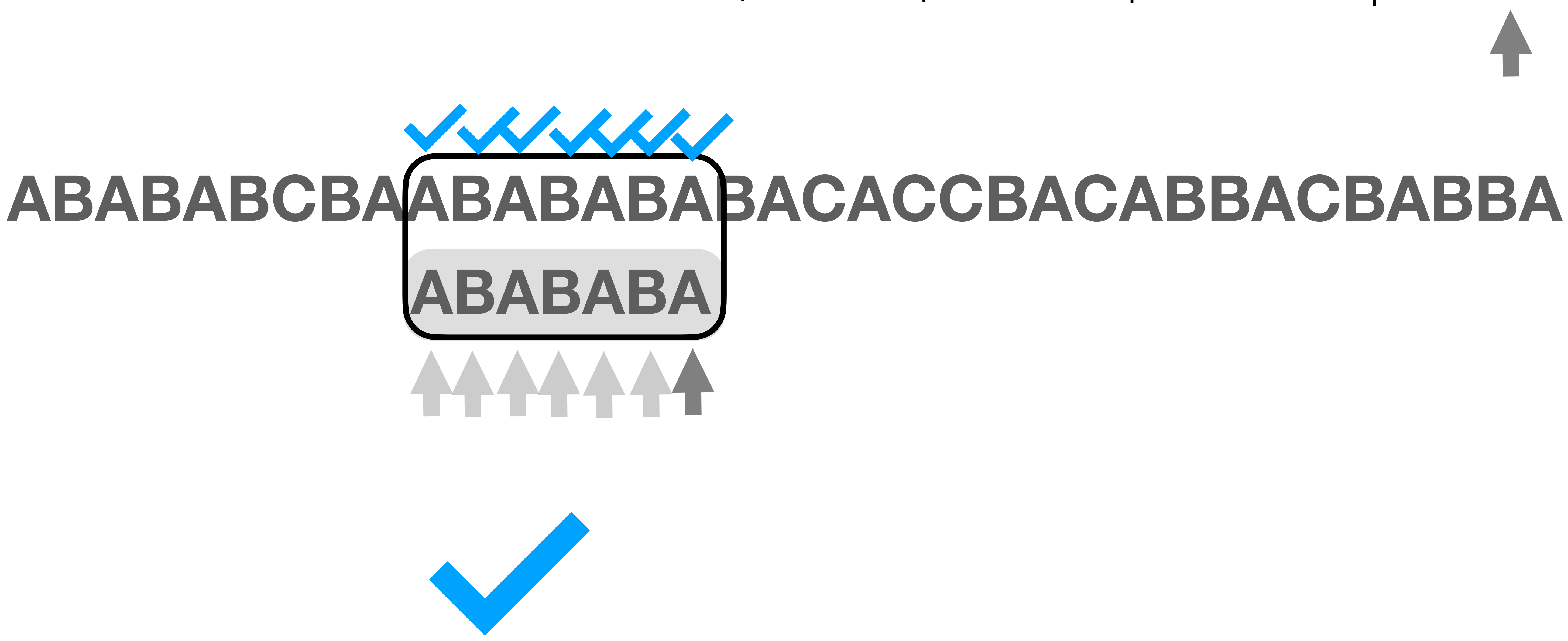
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A (..first 1 char)	AB (..first 2 char)	ABA	ABAB	ABABA	ABABAB (..first 6 char)	ABABABA (..first 7 char)
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

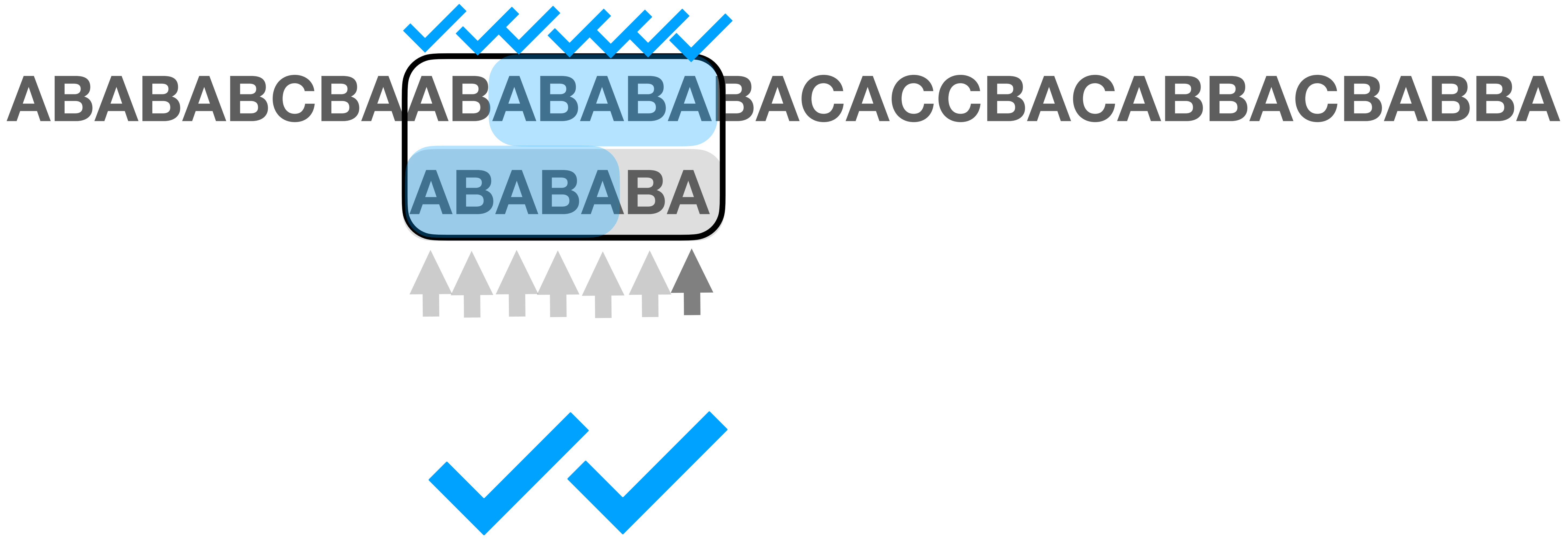
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	0	1	2	3	4	5



Algorithms

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

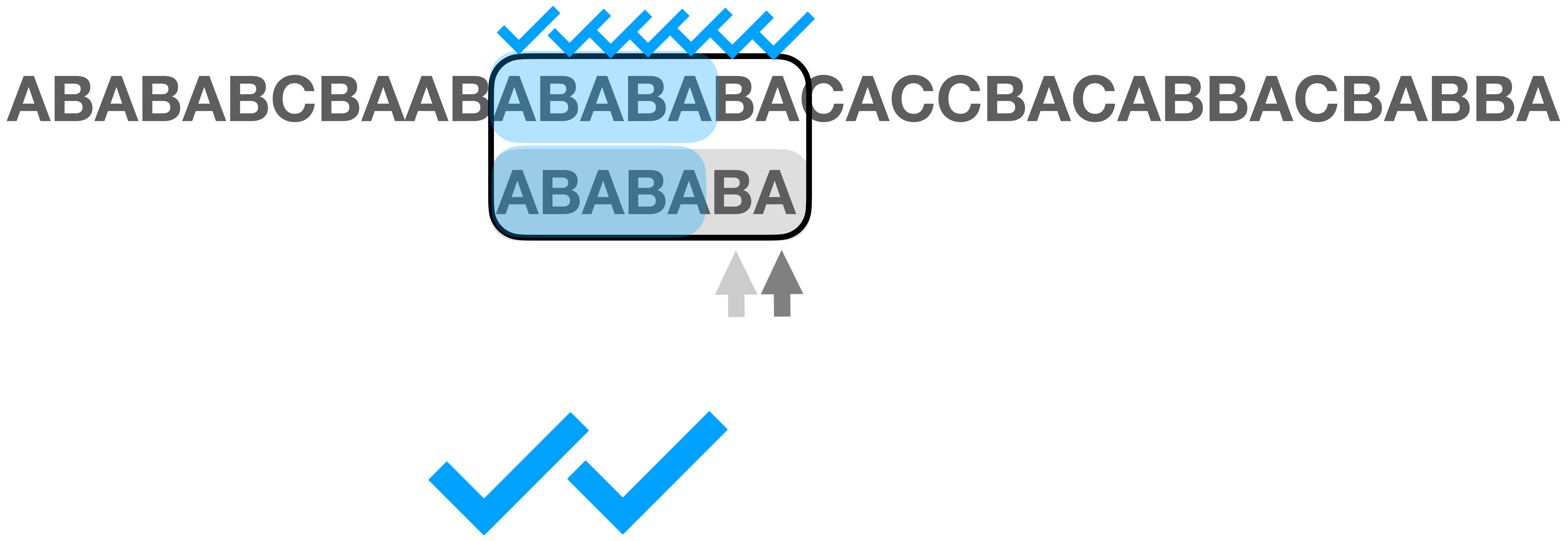
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A (..first 1 char)	AB (..first 2 char)	ABA	ABAB	ABABA	ABABAB (..first 6 char)	ABABABA (..first 7 char)
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

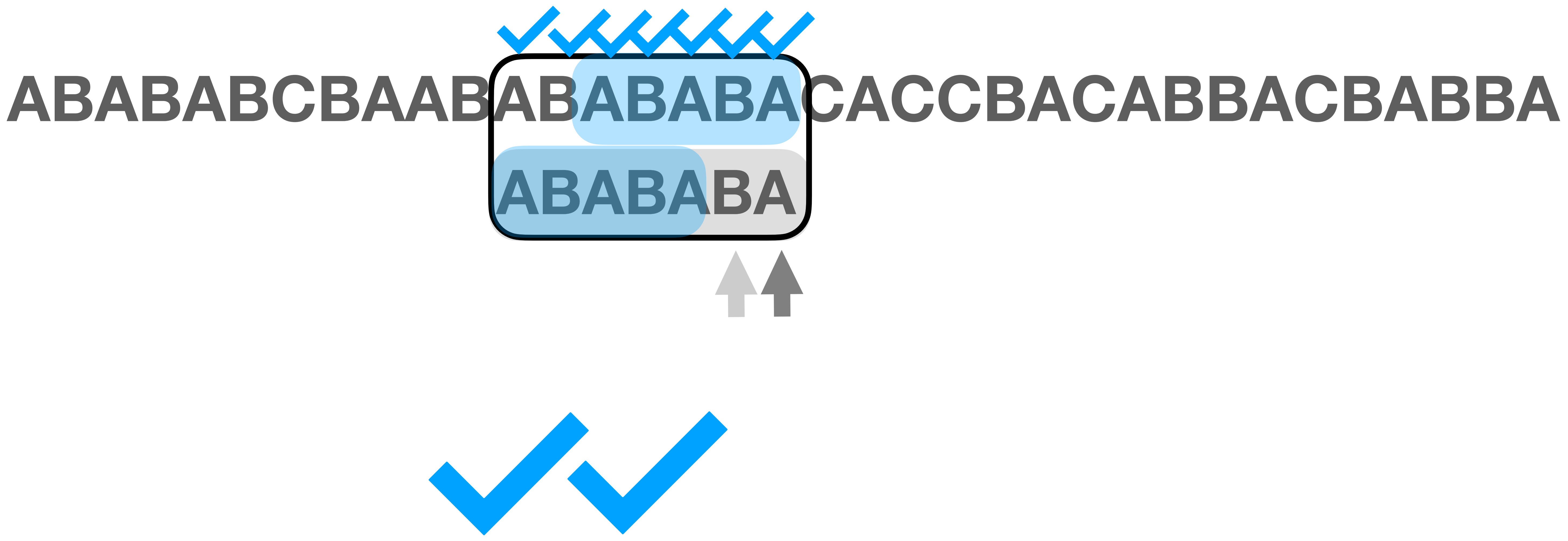
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

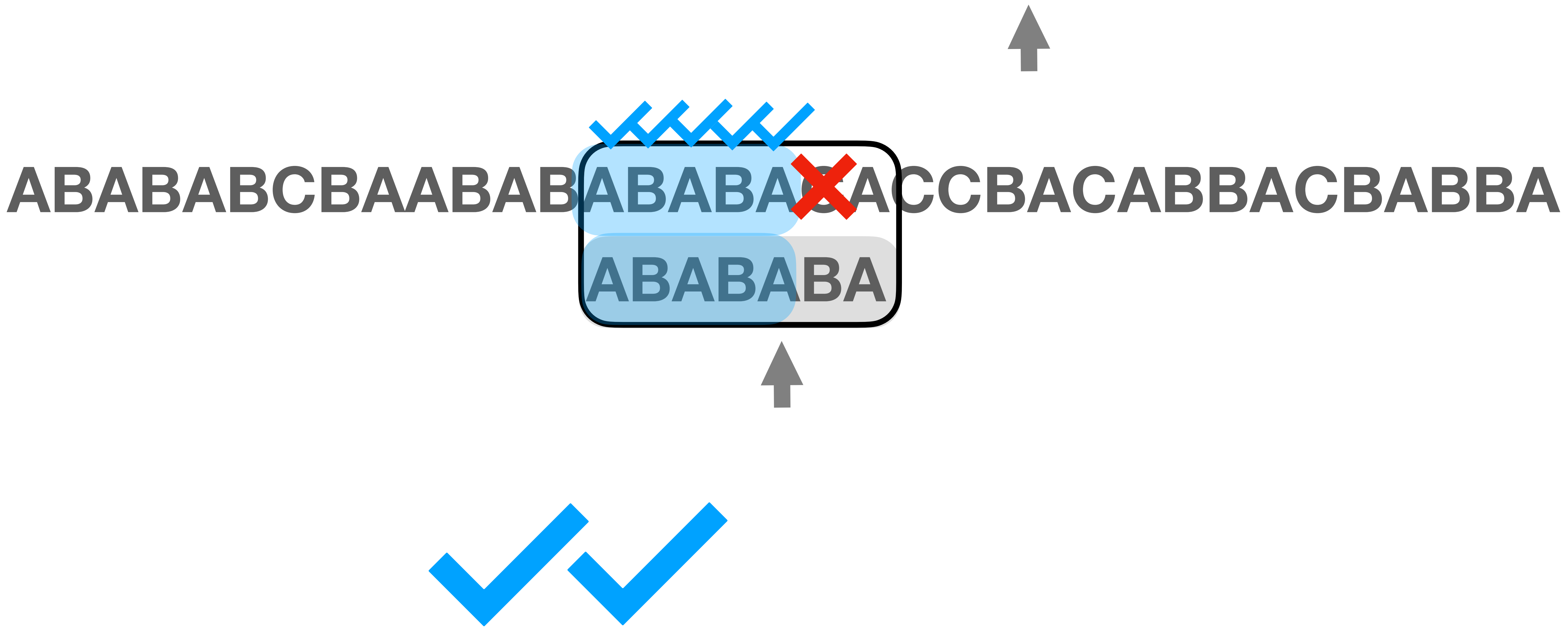
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

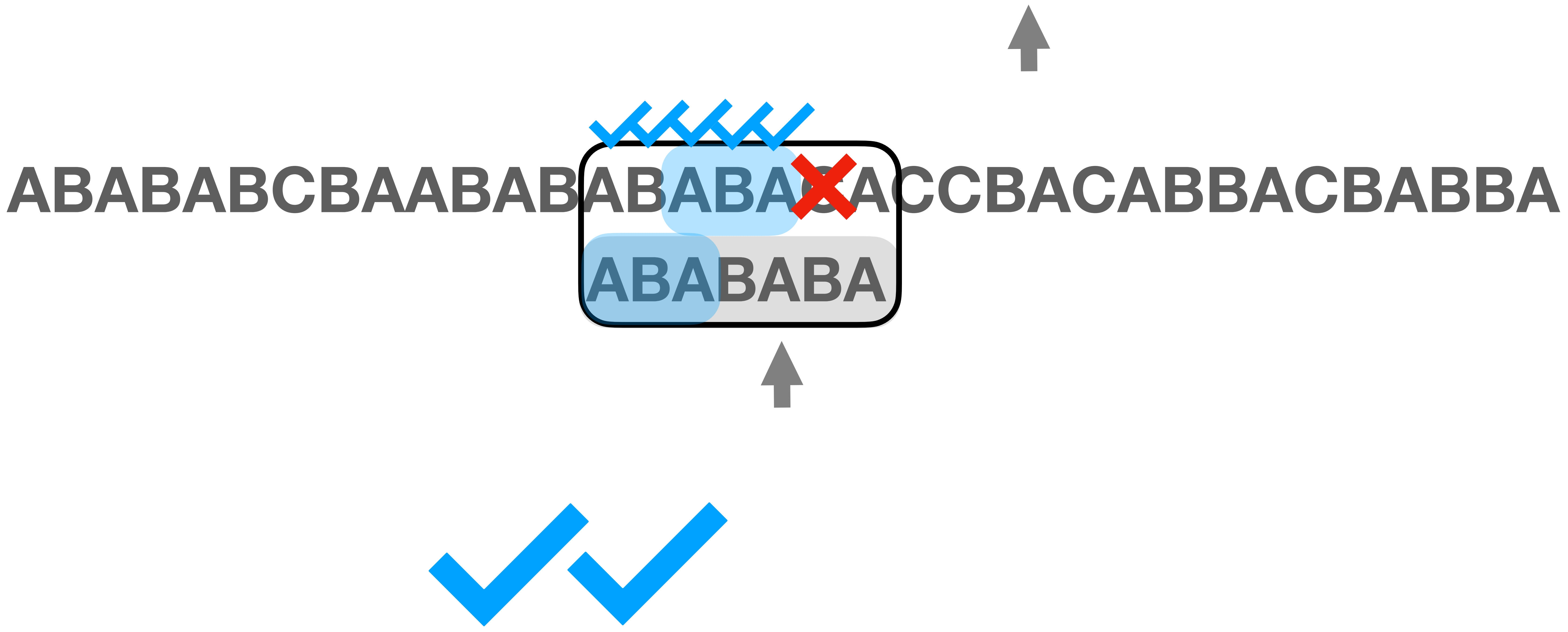
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

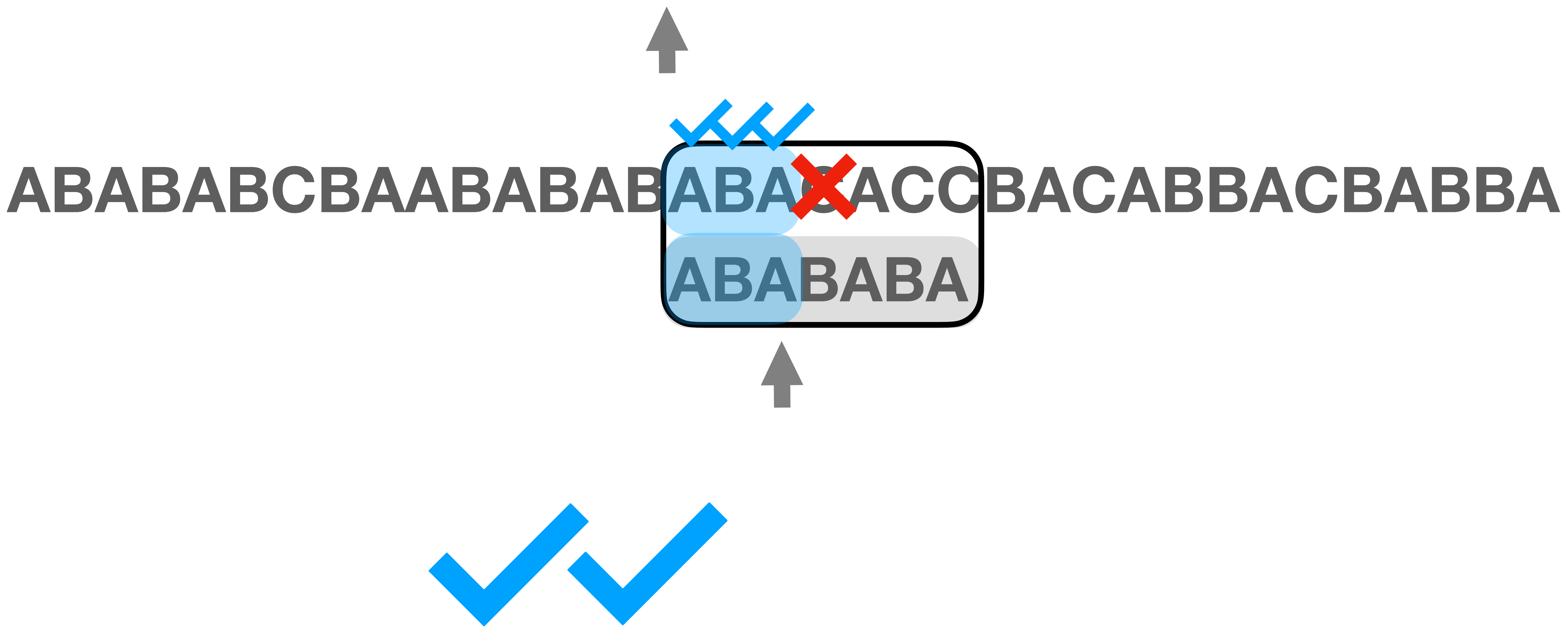
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

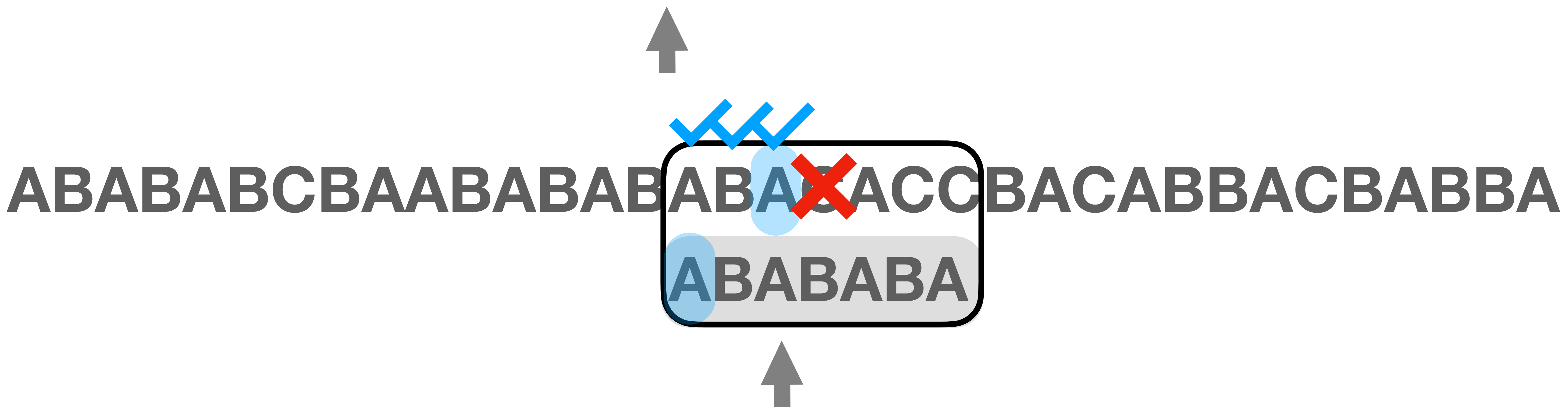
"find a pattern of length m in an array of length n"

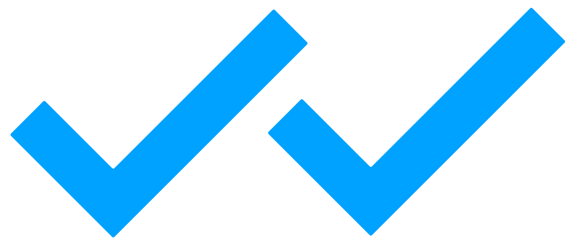
Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5





Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

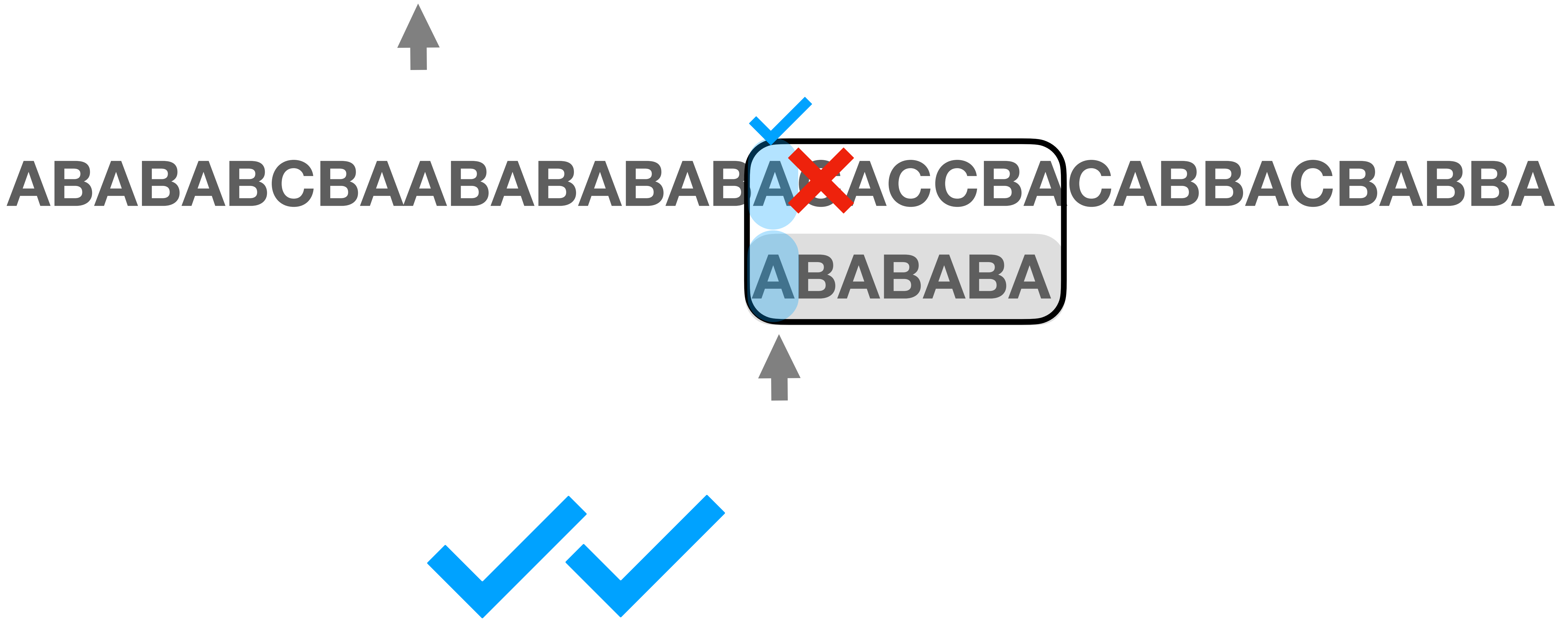
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	**0**	**1**	**2**	**3**	**4**	**5**



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

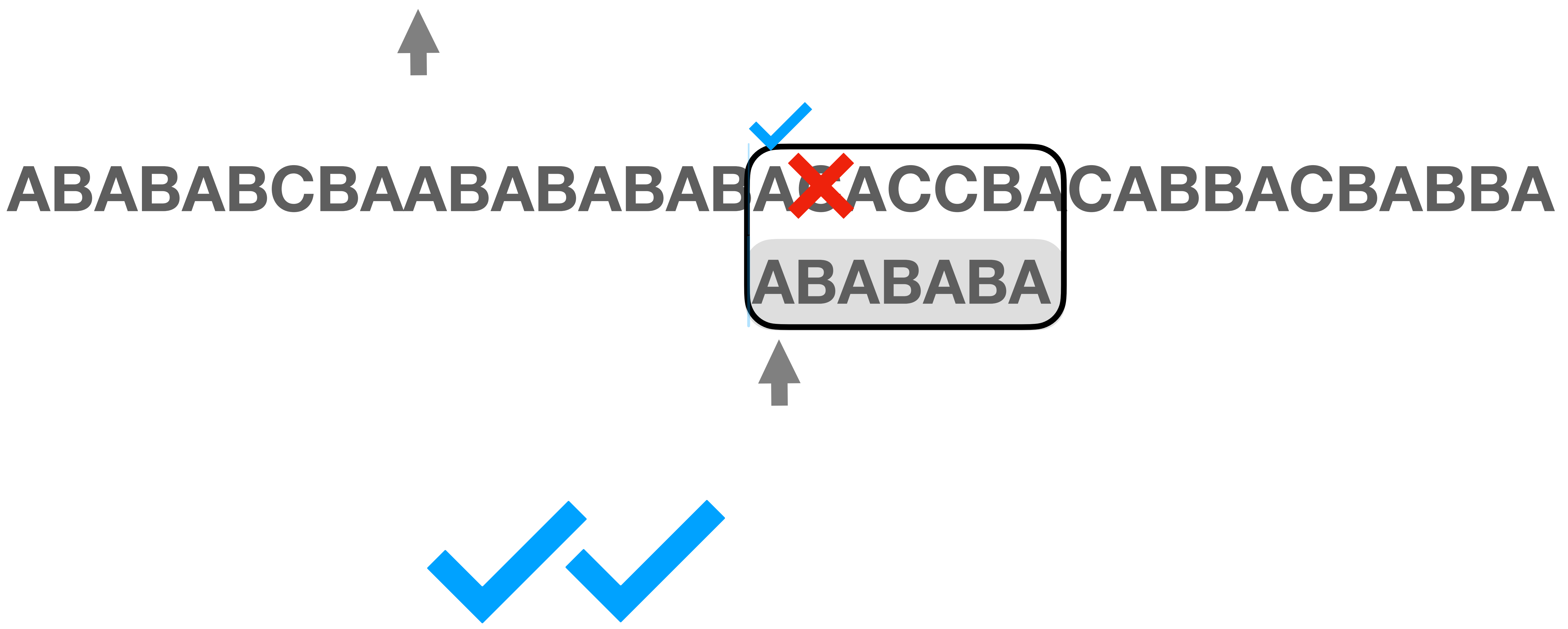
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5

ABABABCBAABABABABACACCBACABBACBABBA

ABABABA

Algorithms

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

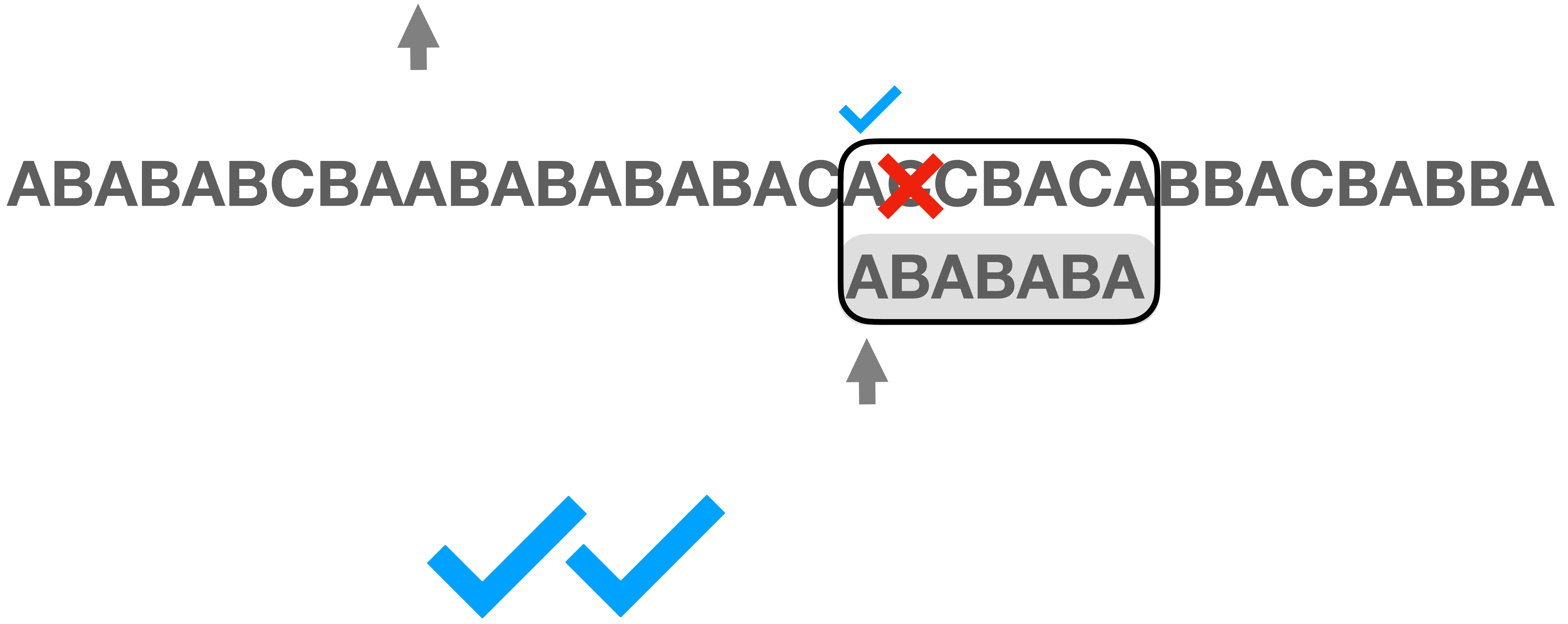
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA**   Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

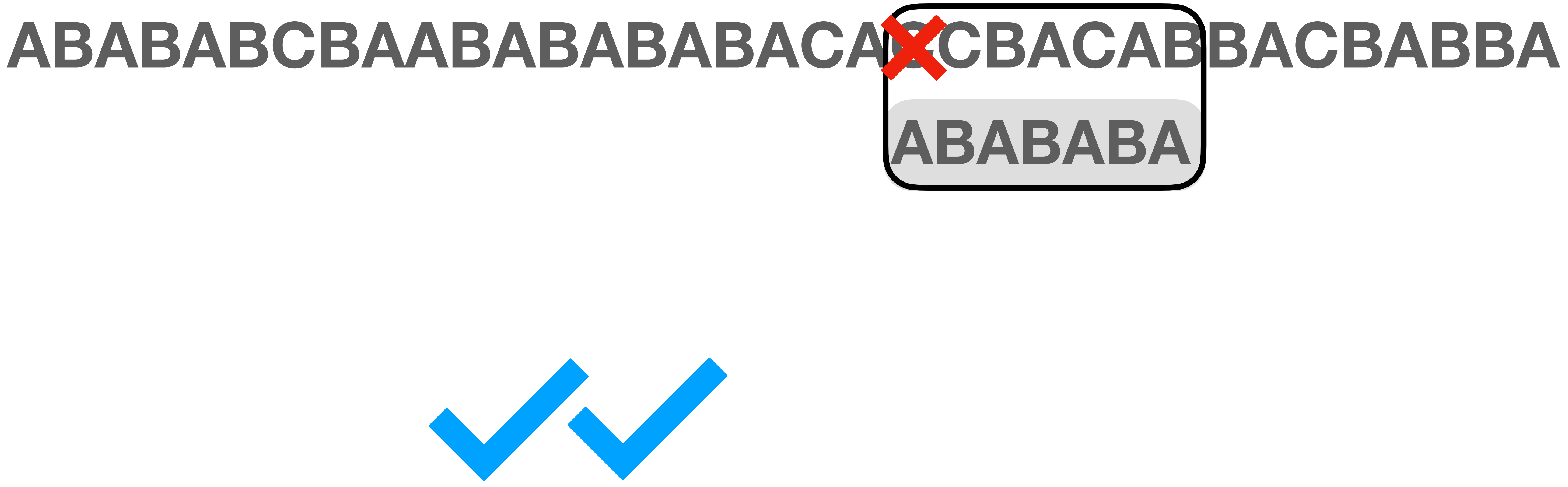
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

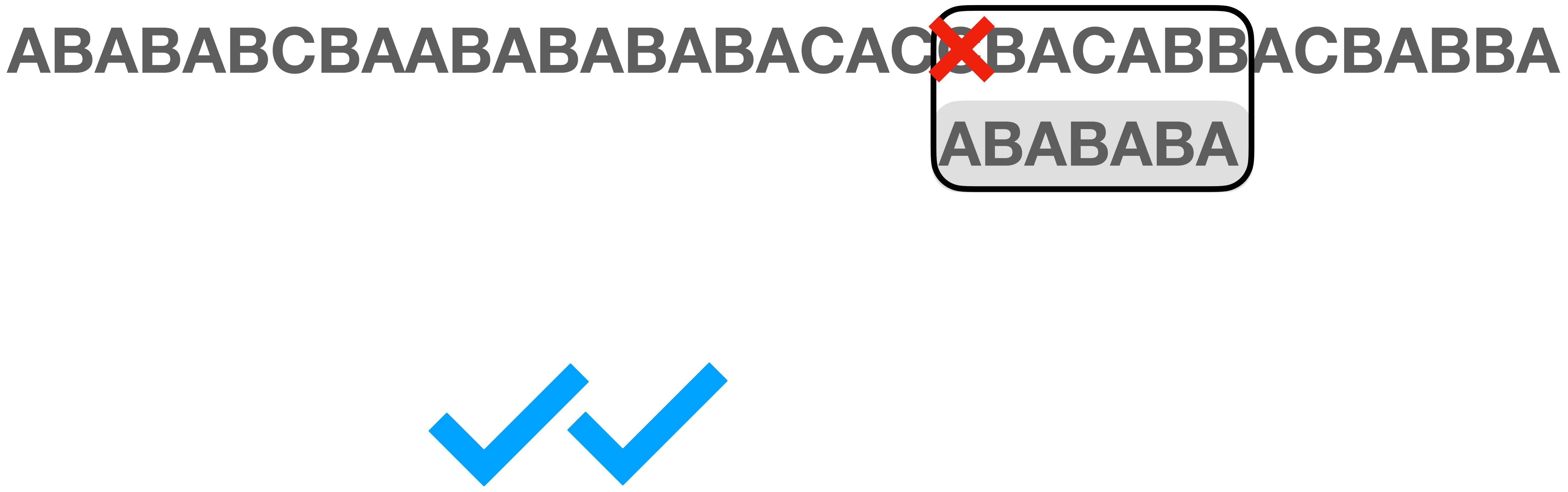
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char A	*..first 2 char* AB	ABA	ABAB	ABABA	*..first 6 char* ABABAB	*..first 7 char* ABABABA
0	0	1	2	3	4	5



Algorithms

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

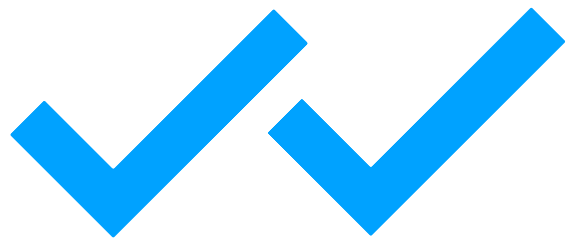
longest proper prefix that is also a suffix"

A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5

(*..first 1 char*, *..first 2 char*, *..first 6 char*, *..first 7 char*)

ABABABCBAABABABABACACCBACABBACBABBA

ABABABA



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Computation Cost: O(n + m)

KMP Algorithm (Knuth-Morris-Pratt)

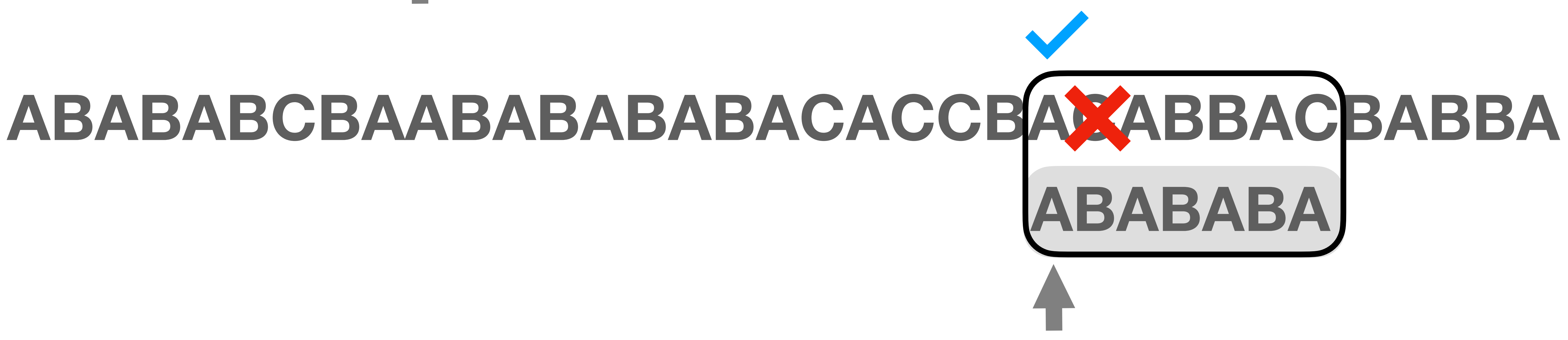
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

"longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	AB	ABA	ABAB	ABABA	ABABAB	ABABABA
0	0	1	2	3	4	5



Algorithms

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

..first 1 char	*..first 2 char*				*..first 6 char*	*..first 7 char*
A	**AB**	**ABA**	**ABAB**	**ABABA**	**ABABAB**	**ABABABA**
0	0	1	2	3	4	5

ABABABCBAABABABABACACCBACABBACBABBA

ABABABA

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

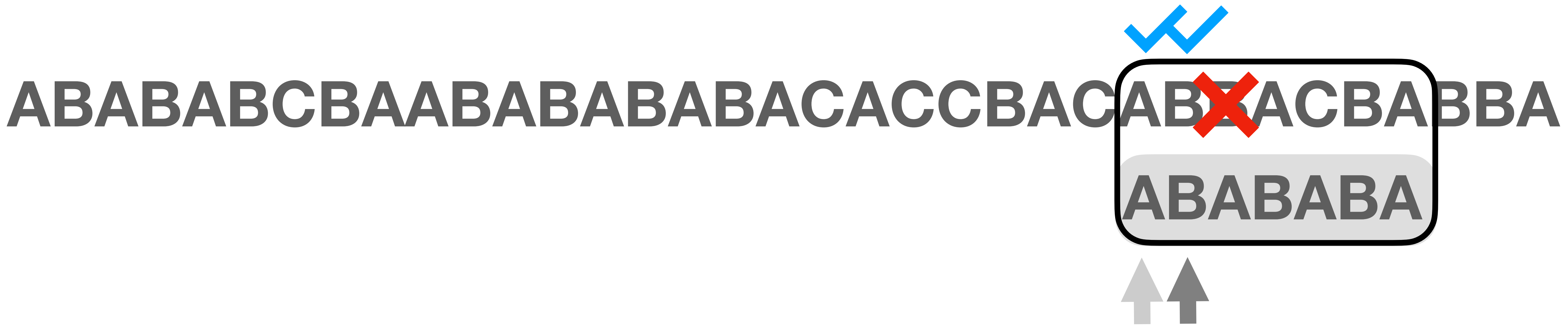
"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

longest proper prefix that is also a suffix"

A *..first 1 char*	AB *..first 2 char*	ABA	ABAB	ABABA	ABABAB *..first 6 char*	ABABABA *..first 7 char*
0	0	1	2	3	4	5



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

KMP Algorithm (Knuth-Morris-Pratt)

Computation Cost: O(n + m)

"find a pattern of length m in an array of length n"

Pattern (m): **ABABABA** Array (n): **ABABABCBAABABABABACACCBACABBACBABBA**

Partial Match Table:

"longest proper prefix that is also a suffix"

A *(..first 1 char)*	AB *(..first 2 char)*	ABA	ABAB	ABABA	ABABAB *(..first 6 char)*	ABABABA *(..first 7 char)*
0	0	1	2	3	4	5

ABABABCBAABABABABACACCBACABBACBABBA

ABABABA

Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

Fill Algorithm

- Flood Fill Algorithm

Sorting Algorithms:

Considerations:

- **Computation time:**
  - **O(1), O(n), O(n^2), O(n log(n)), etc**
  - Worst case? Best case? Average case?
- **Stable sort:**
  Is the order of elements with the same value maintained?
  yes = **stable sort**
- **In place:**
  Is the array sorted in place?
- **Auxiliary storage:**
  How much extra storage is required?

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

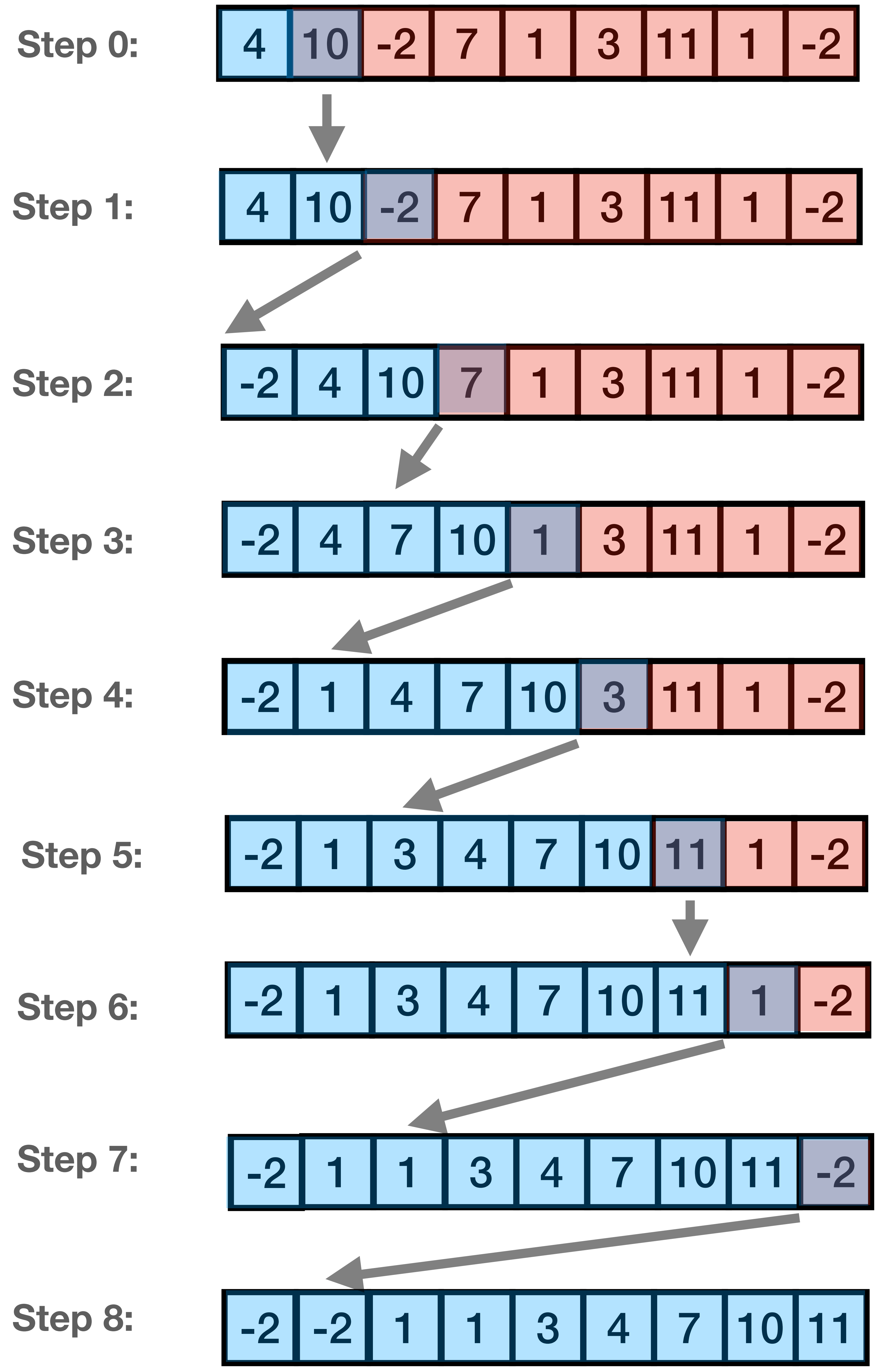
Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Insertion Sort:



Computation Cost:
Worst case: **O(n^2)**
Stable sort: **True**

When to use:
Stable, Quick and dirty

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

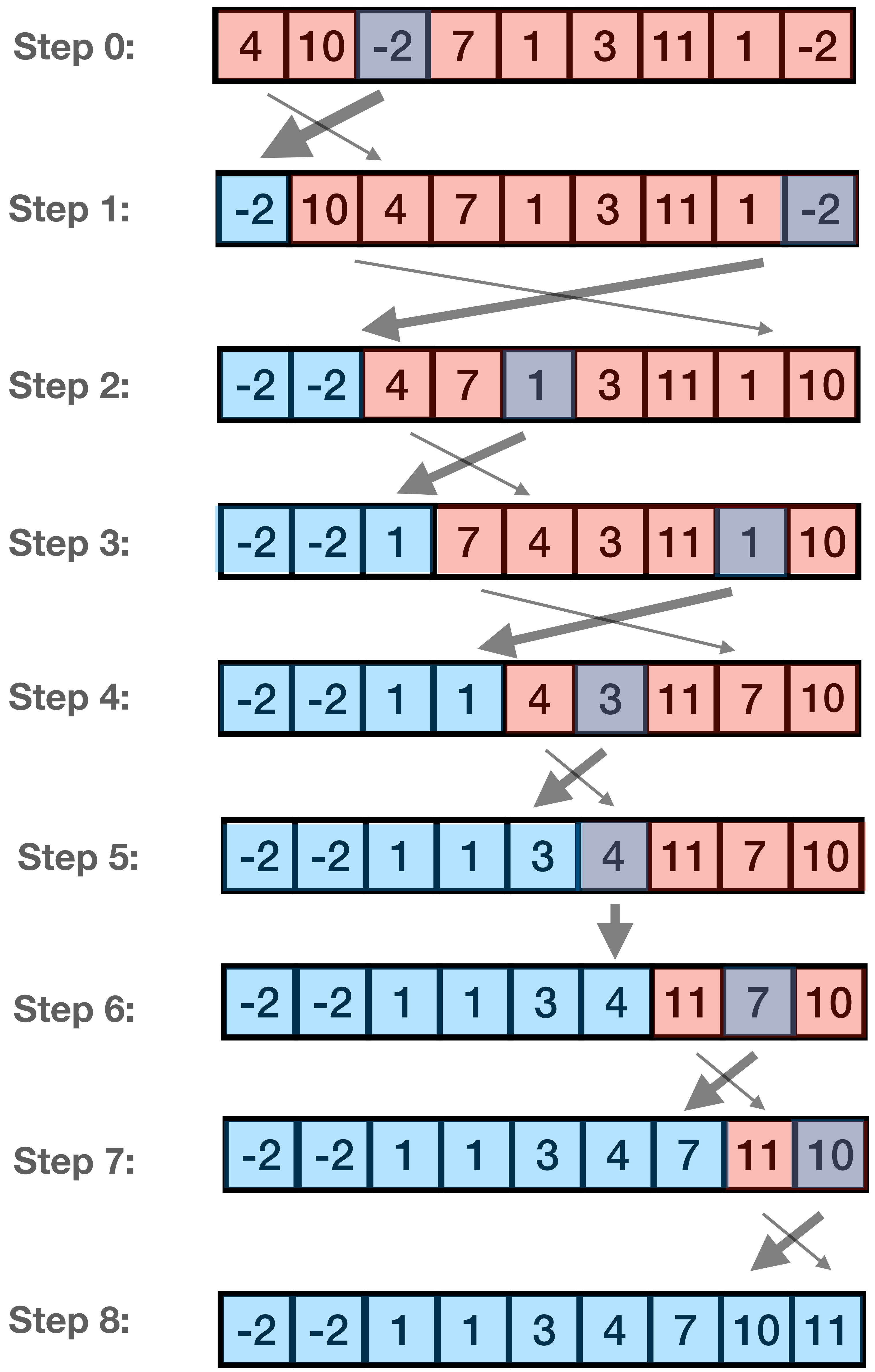
Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Selection Sort:



Computation Cost:
Cost: **O(n^2)**
Stable sort: **False**
In place: **True**

When to use:
Quick and dirty

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

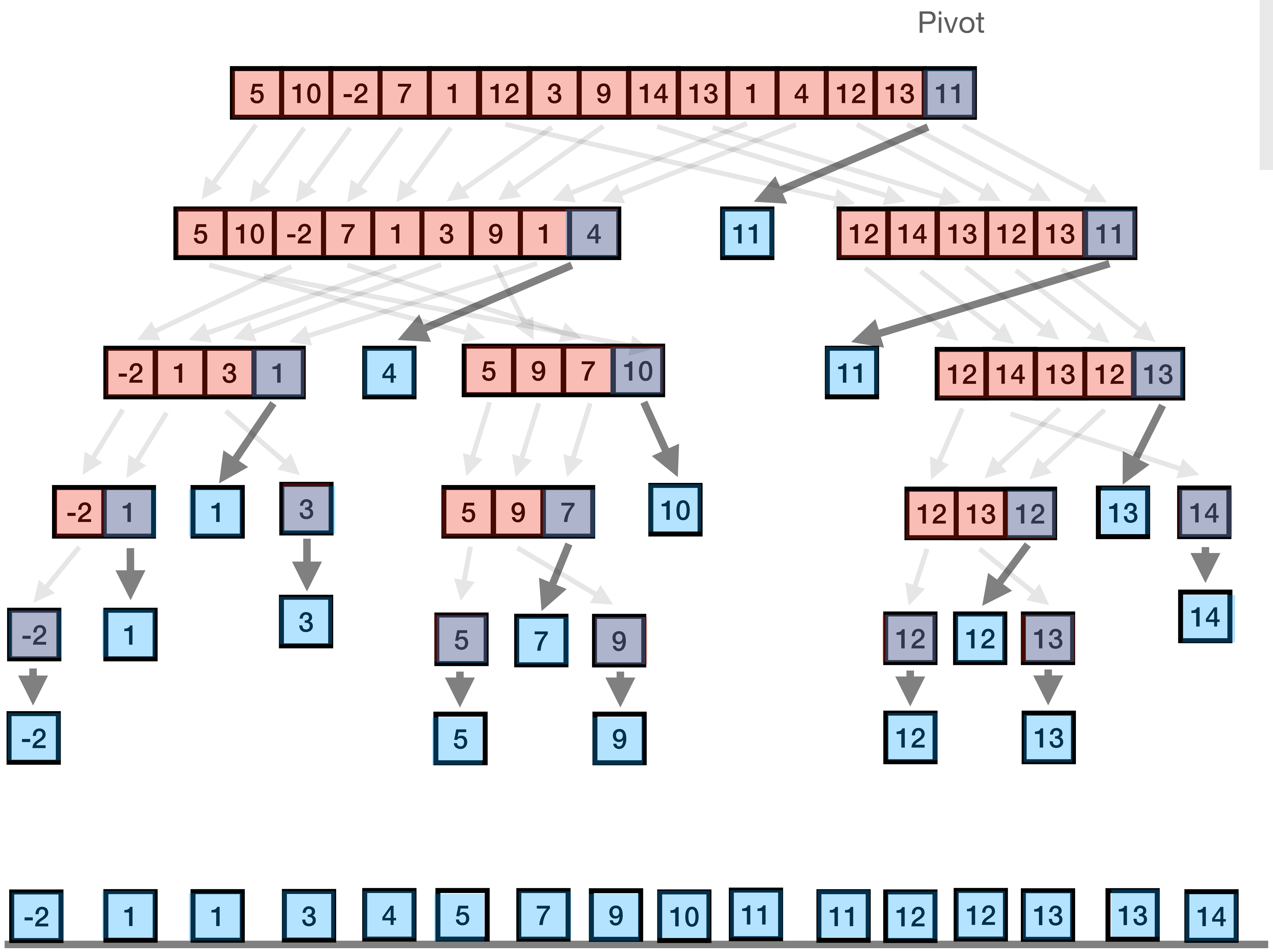
Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Quick Sort:



Computation Cost:
Average: **n log(n)**
Worst case: **n^2 (rare)**
Extra storage: **log(n)**
Stable: **False**
In place: **True**

When to use:
when you don't need a stable sort and average case is more important than worst case. A good implementation uses **O(log(n))** auxiliary storage.

Note:
Best performance happens when arrays split into similar sizes. Lopsided splits cause bad performance.

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

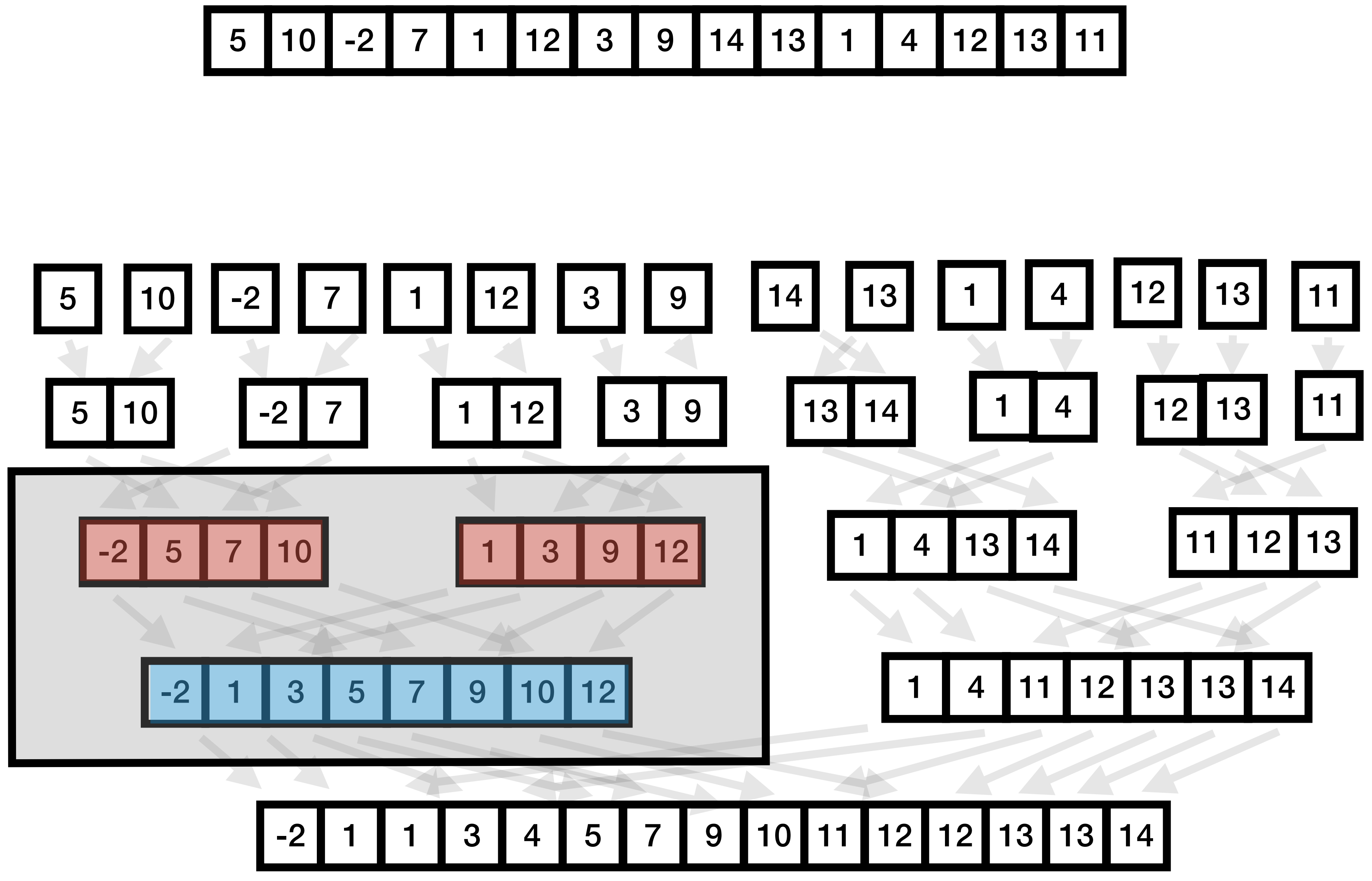
Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

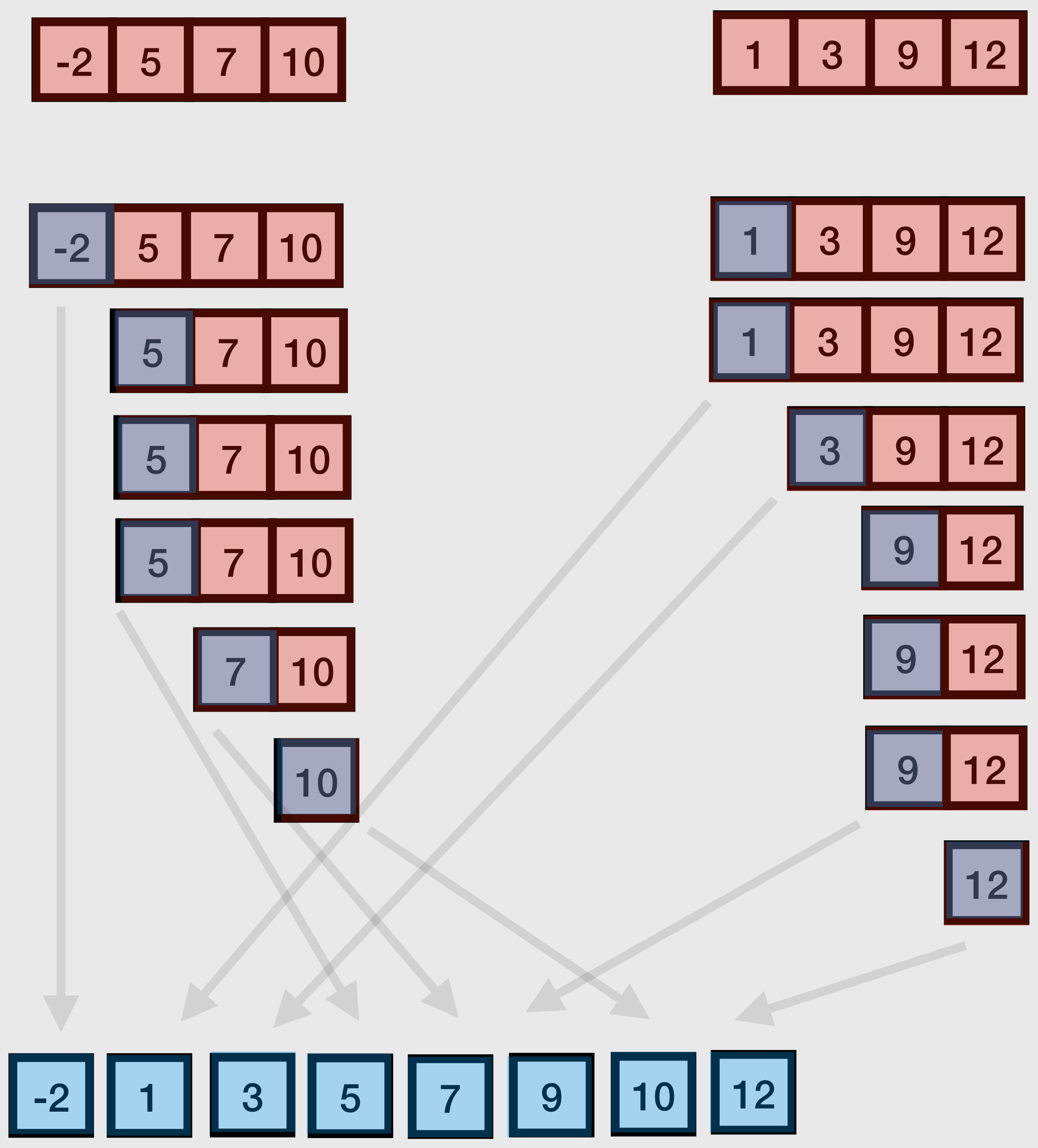
Merge Sort:



Computation Cost:
Worst case: **O(n log(n))**
Extra storage: **O(n)**
Stable: **True**
In place: **sub-optimal**

When to use:
when you need a stable **O(nlog(n))** sort. A downside is that it uses **O(n)** auxiliary space. In place options are suboptimal.

Merge Substep:



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

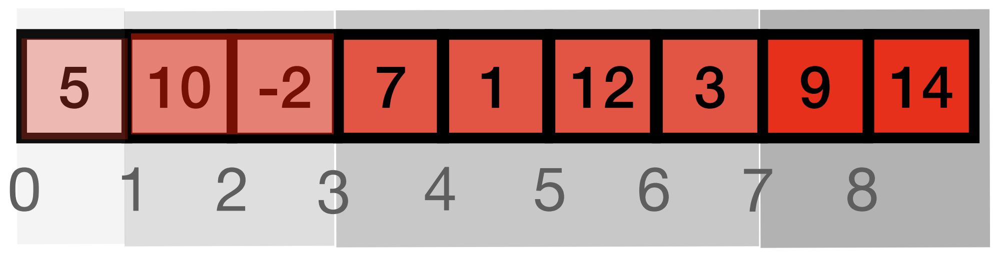
Heap Sort:

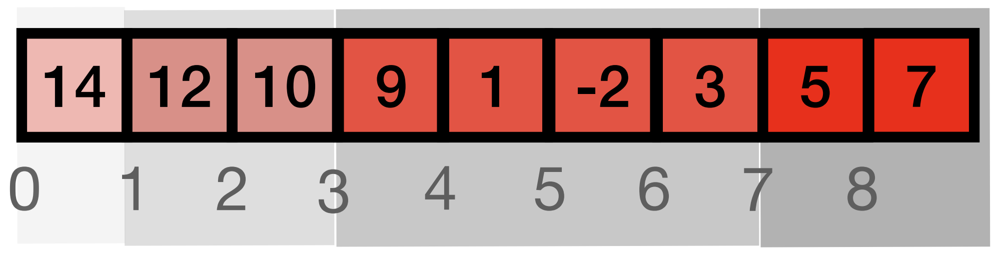
5	10	-2	7	1	12	3	9	14

Computation Cost:
Worst case: **O(nlog(n))**
Extra storage: **O(1)**
In place: **possible**
Stable: **False**

Uses a data structure called a heap…

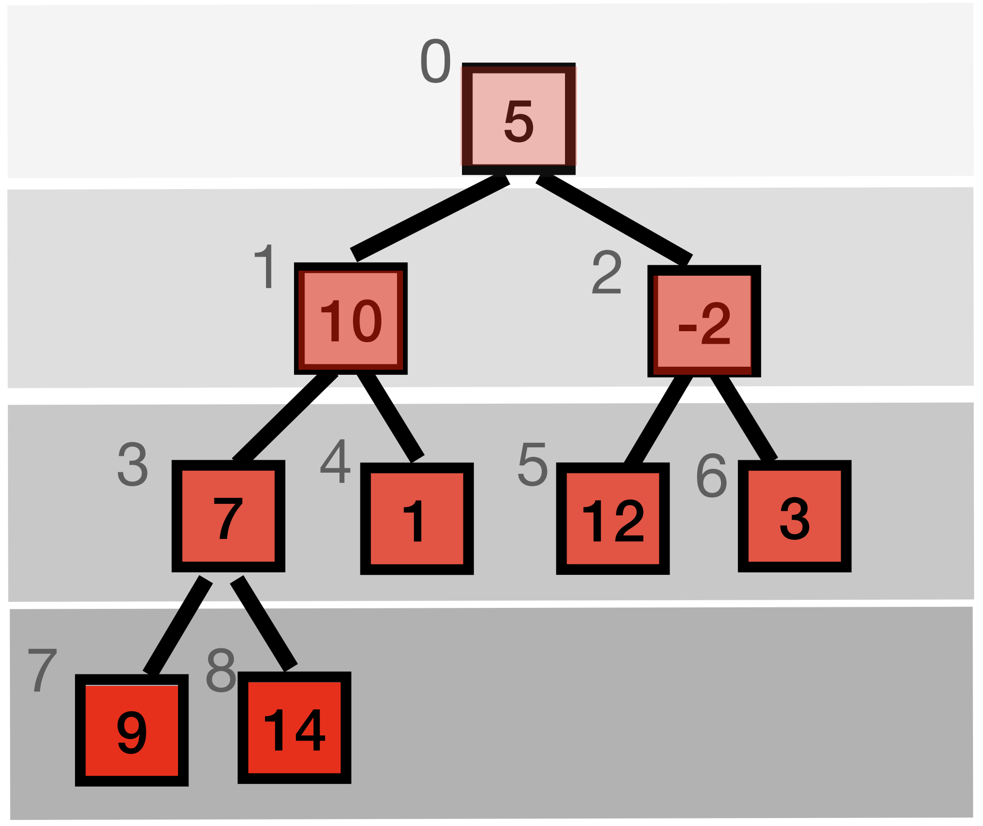
index:



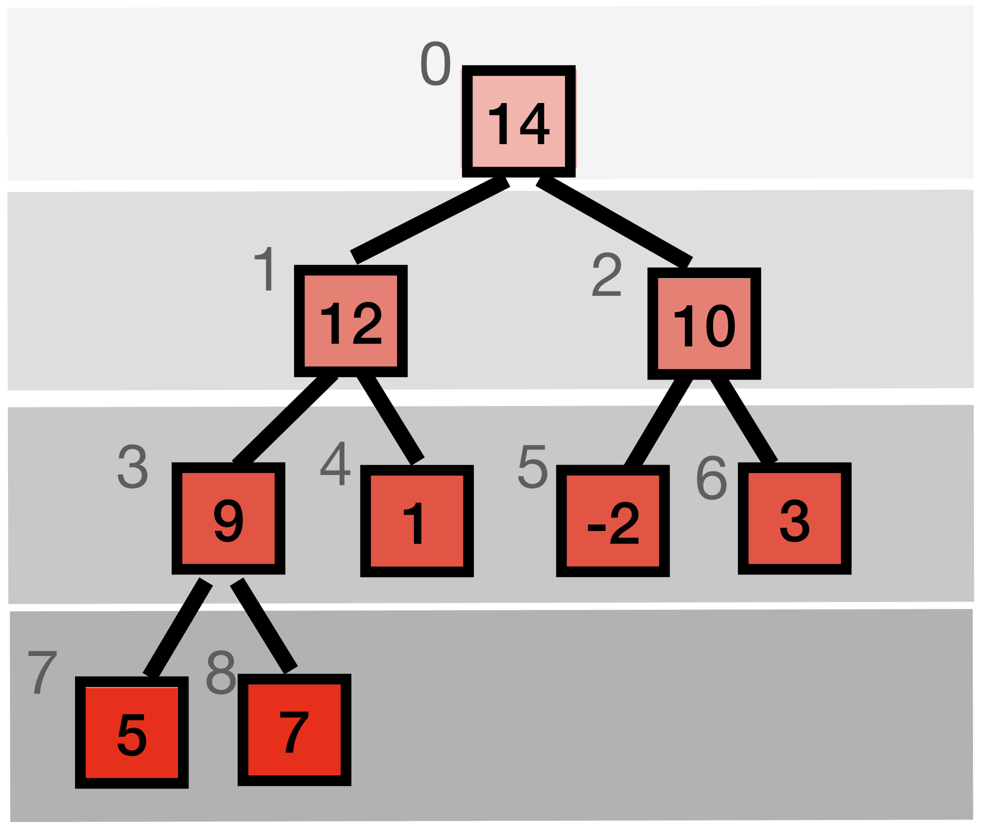


- usually slower than quicksort
- better worst case performance

Heap



Max Heap



When to use:
when you don't need a stable sort and you care more about worst case performance than average. Also uses constant auxiliary storage space.

Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

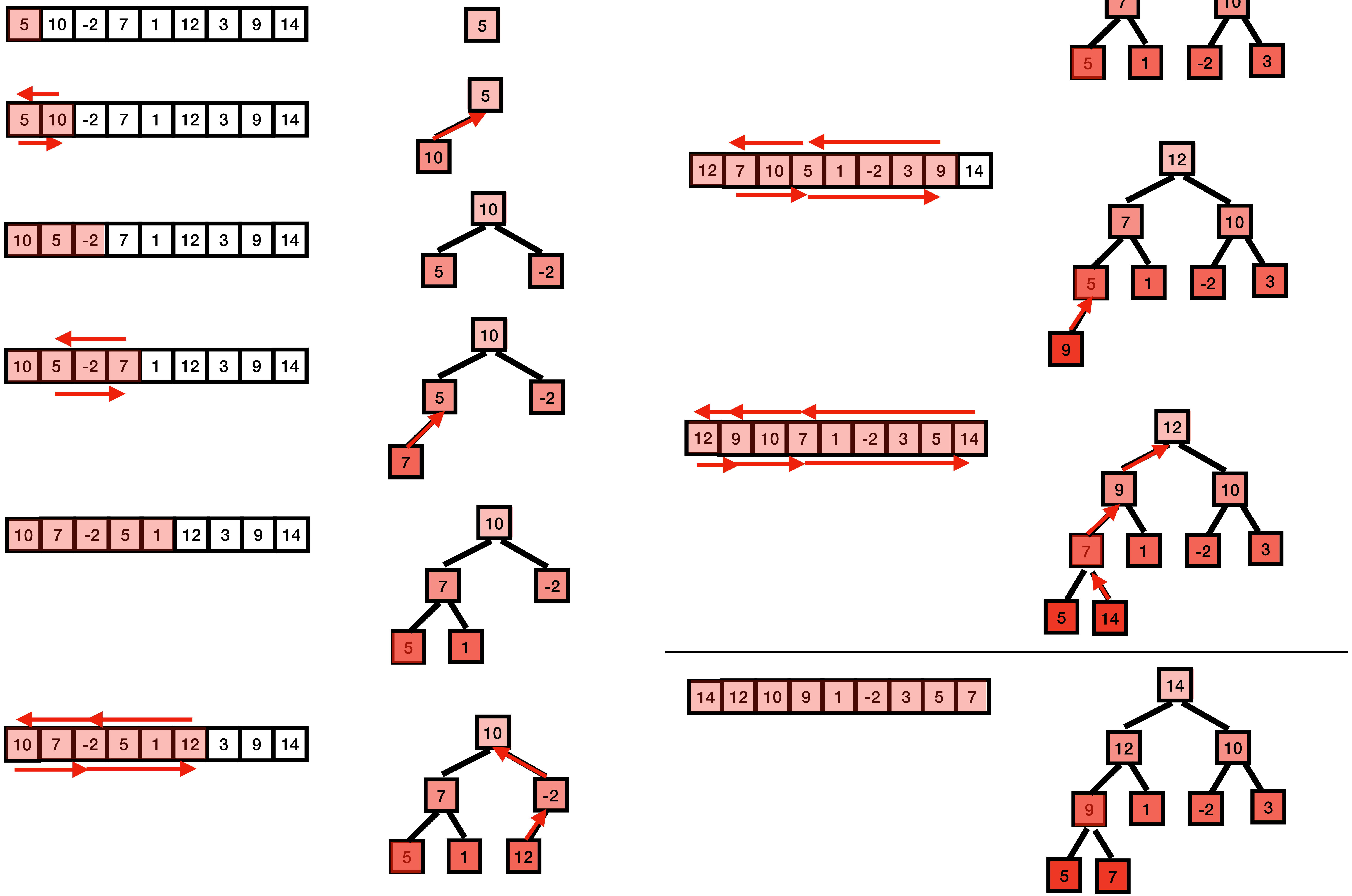
Fill Algorithm

- Flood Fill Algorithm

Heap Sort:

5	10	-2	7	1	12	3	9	14

… build max heap



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

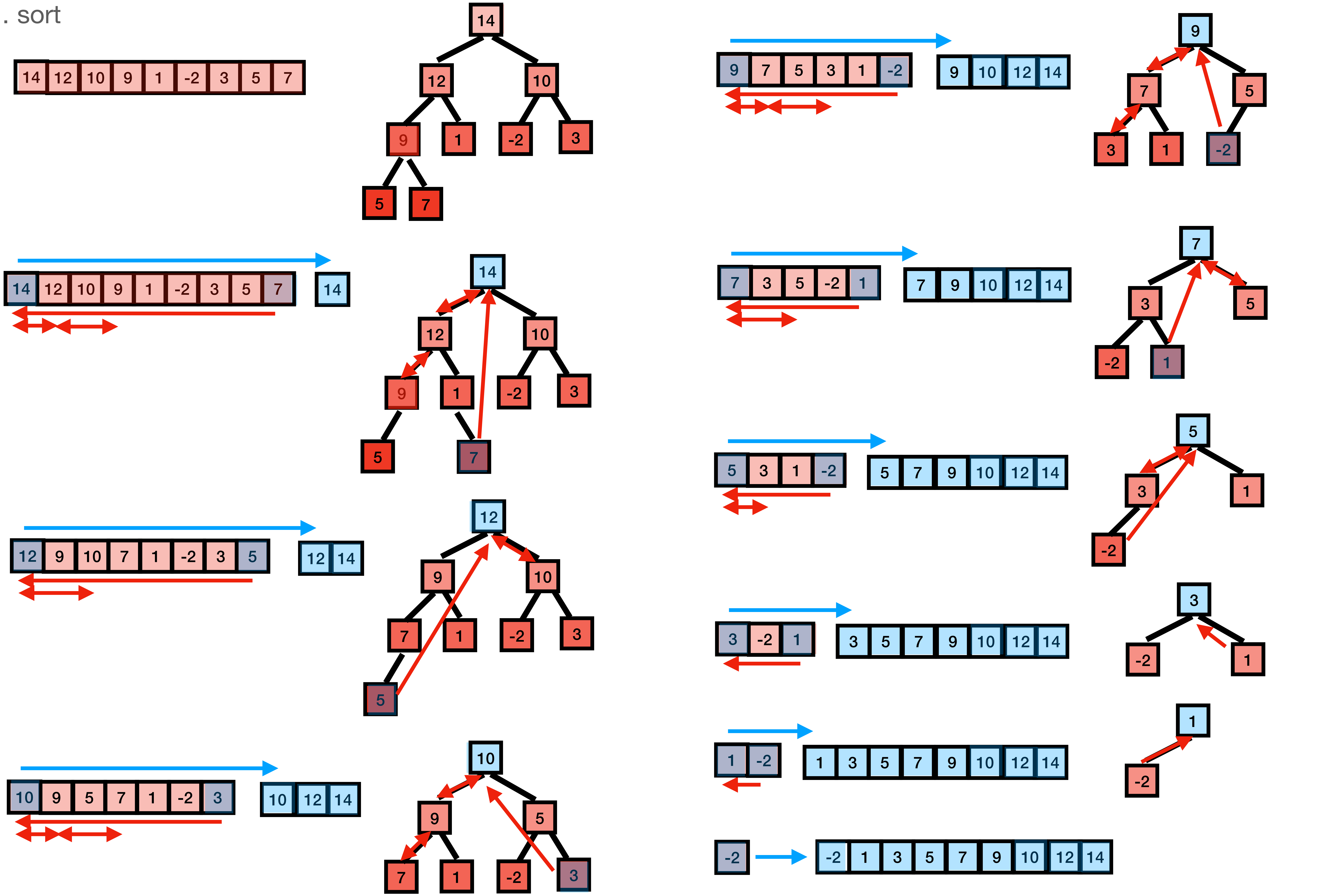
Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Heap Sort:

5	10	-2	7	1	12	3	9	14

… sort



Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

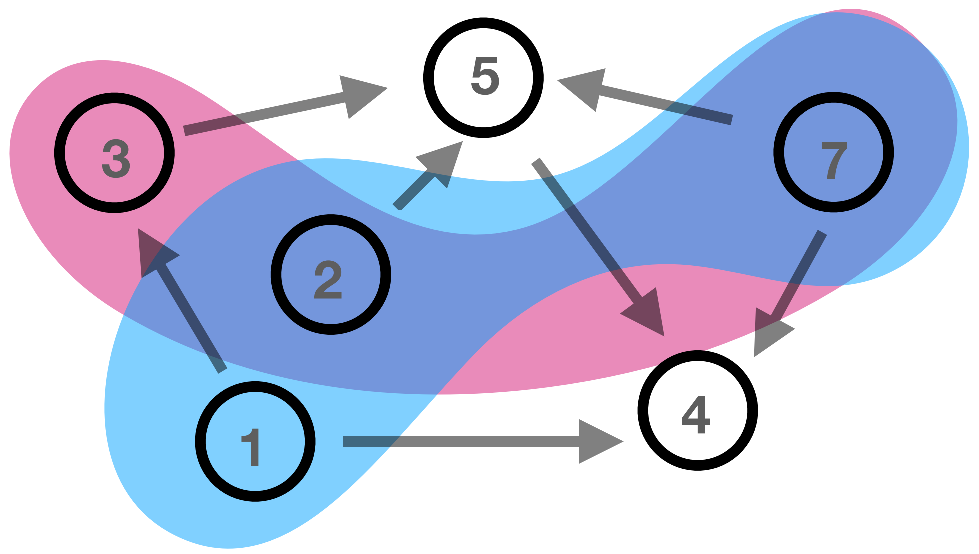
Compression Algorithms

- Huffman Coding

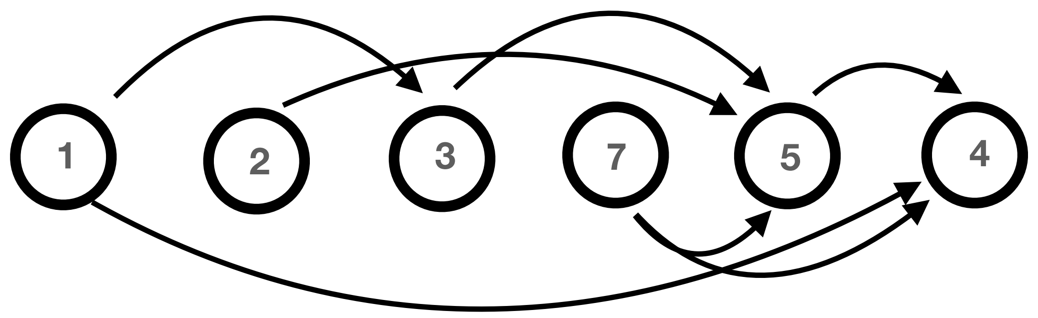
Fill Algorithm

- Flood Fill Algorithm

Kahn's Topological Sort



Computation Cost:
Cost: **O(|V|+|E|))**



Directed Acyclic Graph (DAG)

…sort in order consistent with graph

many consistent sorting orders…

Options:

	1, 2, 7	**any order to start…**
if 1 before 3	3, 2, 7	**any order after 1…**
	5	**next to last**
	4	**last**

Acceptable sorts…

1, 2, 7, 3, 5, 4
1, 7, 2, 3, 5, 4
1, 3, 2, 7, 5, 4
2, 1, 3, 7, 5, 4
7, 2, 1, 3, 5, 4
⋮

Idea/Pseudo-code:

```
Initialize:
S = [ ];   sorted vertices
V = [ ];   vertices with no incoming edges

While S is non-empty:
    Move a vertex u from V to end of S
    For each vertex v with edge e from u to v
        Remove edge e from graph
        If v has no other incoming edges
            Add v into V

If any edges left in graph
    "Graph has at least one cycle"
else:
    "S is a topological sort."
```

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

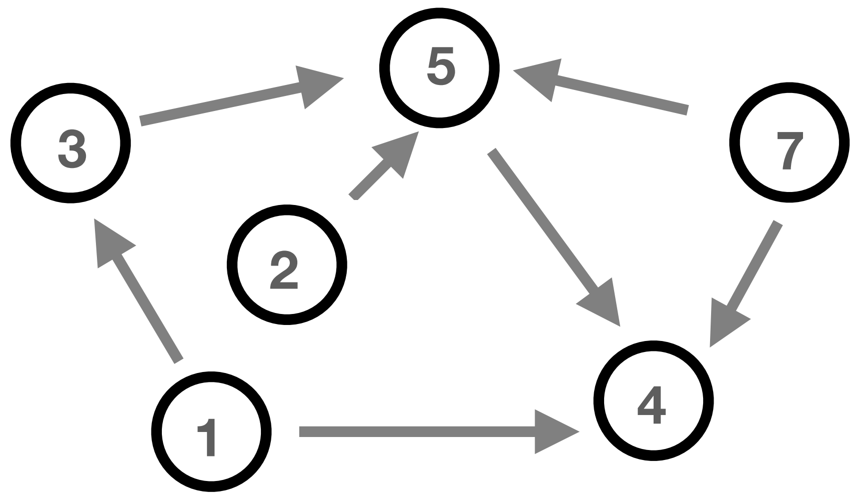
Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

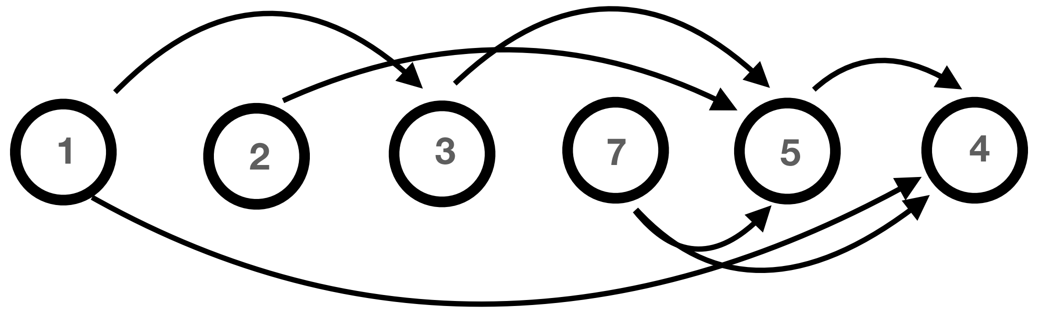
Kahn's Topological Sort

Directed Acyclic Graph (DAG)

…sort in order consistent with graph



Computation Cost:
Cost: **O(|V|+|E|))**



many consistent sorting orders…

Options:

	1, 2, 7	**any order to start…**
if 1 before 3	3, 2, 7	**any order after 1…**
	5	**next to last**
	4	**last**

Idea/Pseudo-code:

Initialize:

S = [ ]; sorted vertices
V = [ ]; vertices with no incoming edges

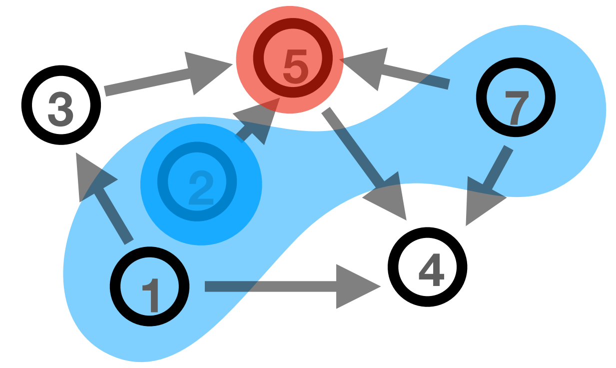
```
While S is non-empty:
    Move a vertex u from V to end of S
    For each vertex v with edge e from u to v
        Remove edge e from graph
        If v has no other incoming edges
            Add v into V

If any edges left in graph
    "Graph has at least one cycle"
else:
    "S is a topological sort."
```

Algorithm:

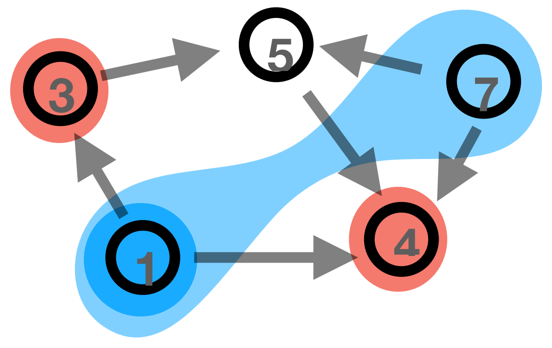
S = [ ]; V = [2, 1 , 7]; pick 2…

… move 2 to S and remove edge to 5
… (5 has other incoming edges)



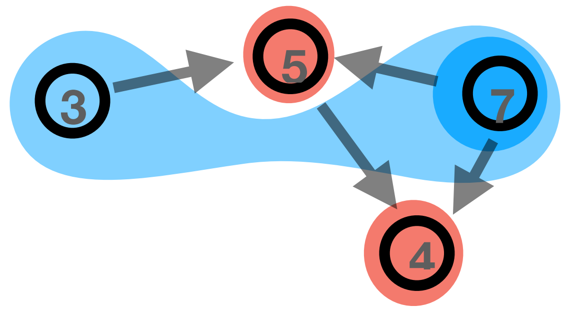
S = [2]; V = [1 , 7]; pick 1…

… move 1 to S and remove edges to 3 and 4
… add 3 to V, (4 has other incoming edges)



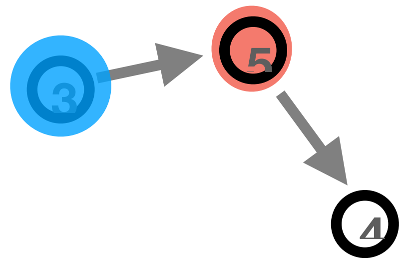
S = [2,1]; V = [7, 3]; pick 7…

… move 7 to S and remove edges to 4 and 5
… (4 and 5 have other incoming edges)



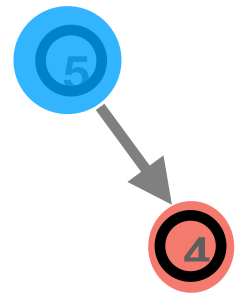
S = [2,1,7]; V = [3]; pick 3…

… move 3 to S and remove edge to 5
… add 5 to V



S = [2,1,7,3]; V = [5]; pick 5…

… move 5 to S and remove edge to 4
… add 4 to V



S = [2,1,7,3,5]; V = [4]; pick 4…

Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

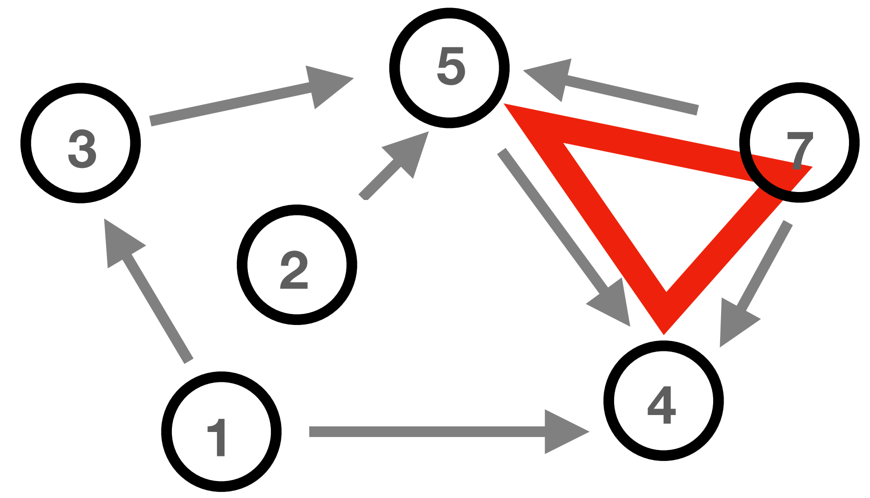
Fill Algorithm

- Flood Fill Algorithm

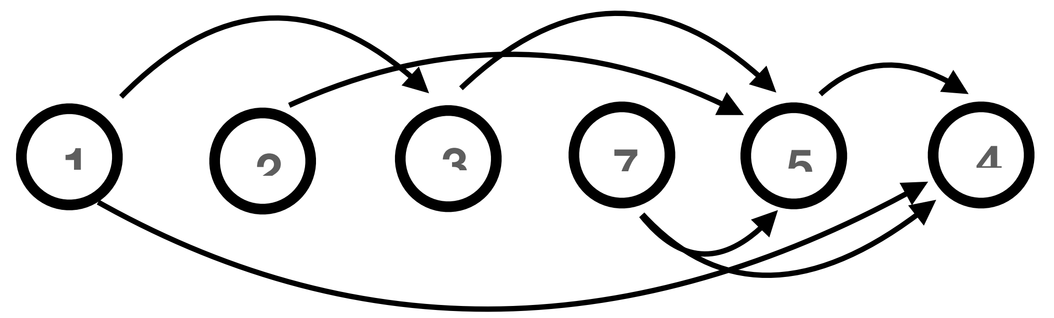
Kahn's Topological Sort

Directed Acyclic Graph (DAG)

…sort in order consistent with graph



Computation Cost:
Cost: **O(|V|+|E|))**



many consistent sorting orders…

Options:

	1, 2, 7	**any order to start…**
if 1 before 3	3, 2, 7	**any order after 1…**
	5	**next to last**
	4	**last**

Idea/Pseudo-code:

Initialize:

S = [ ]; sorted vertices
V = [ ]; vertices with no incoming edges

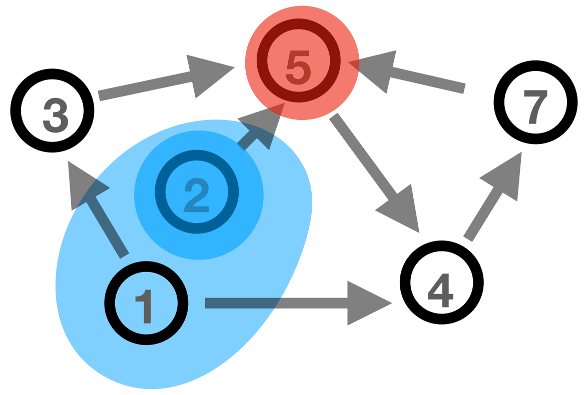
```
While V is non-empty:
    Move a vertex u from V to end of S
    For each vertex v with edge e from u to v
        Remove edge e from graph
        If v has no other incoming edges
            Add v into V

If any edges left in graph
    "Graph has at least one cycle"
else:
    "S is a topological sort."
```

Algorithm:

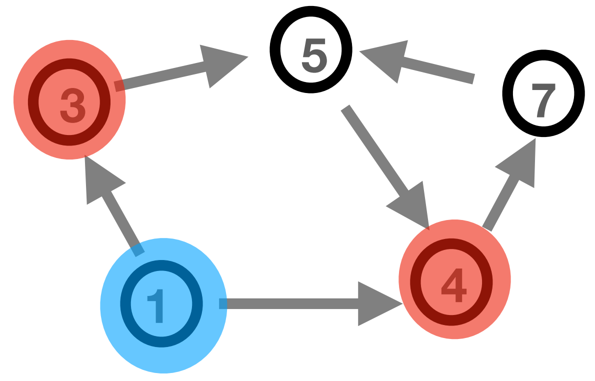
S = [ ]; V = [2, 1]; pick 2…

… move 2 to S and remove edge to 5
… (5 has other incoming edges)



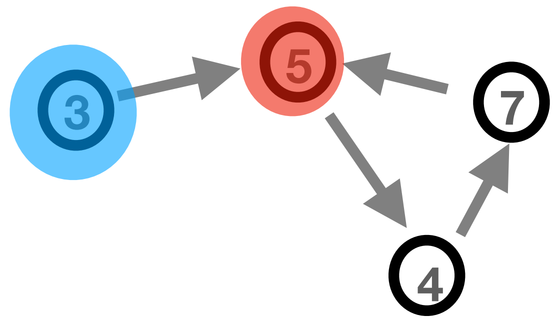
S = [2]; V = [1]; pick 1…

… move 1 to S and remove edges to 3 and 4
… add 3 to V, (4 has other incoming edges)

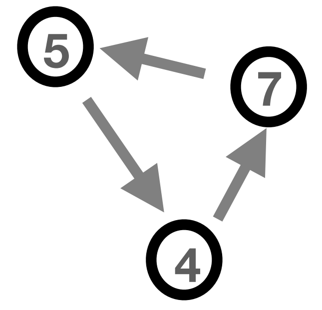


S = [2,1]; V = [3]; pick 3…

… move 3 to S and remove edge to 5
… (5 have other incoming edges)



S = [2,1,3]; V = [ ];



V empty but still edges left…

Graph contains a directed cycle.

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

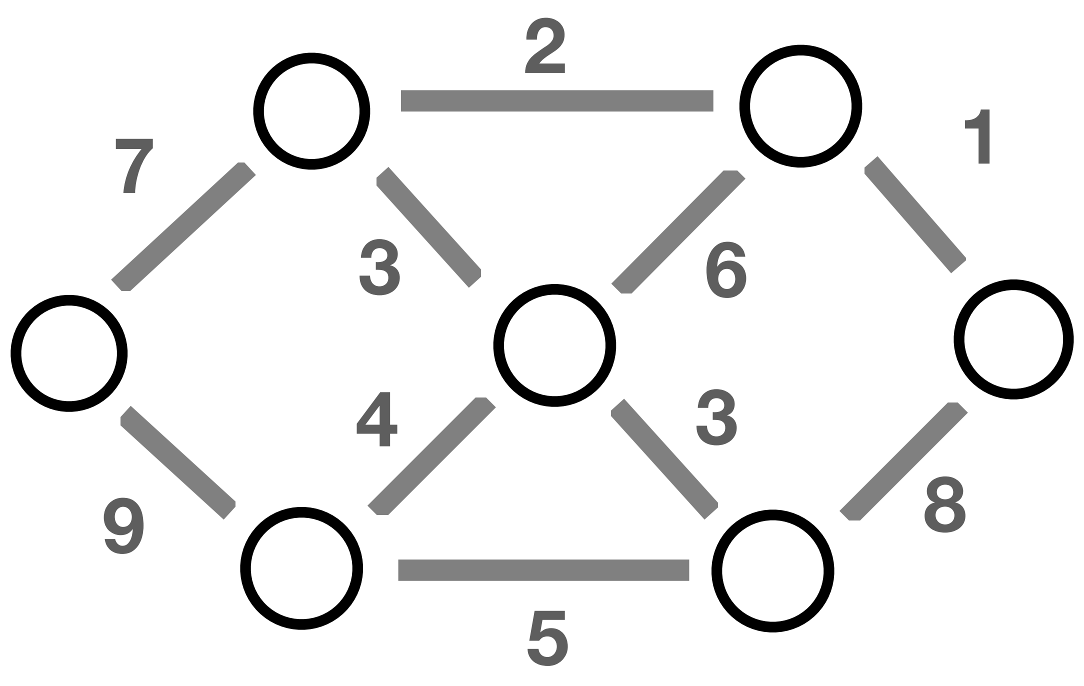
Compression Algorithms
- Huffman Coding

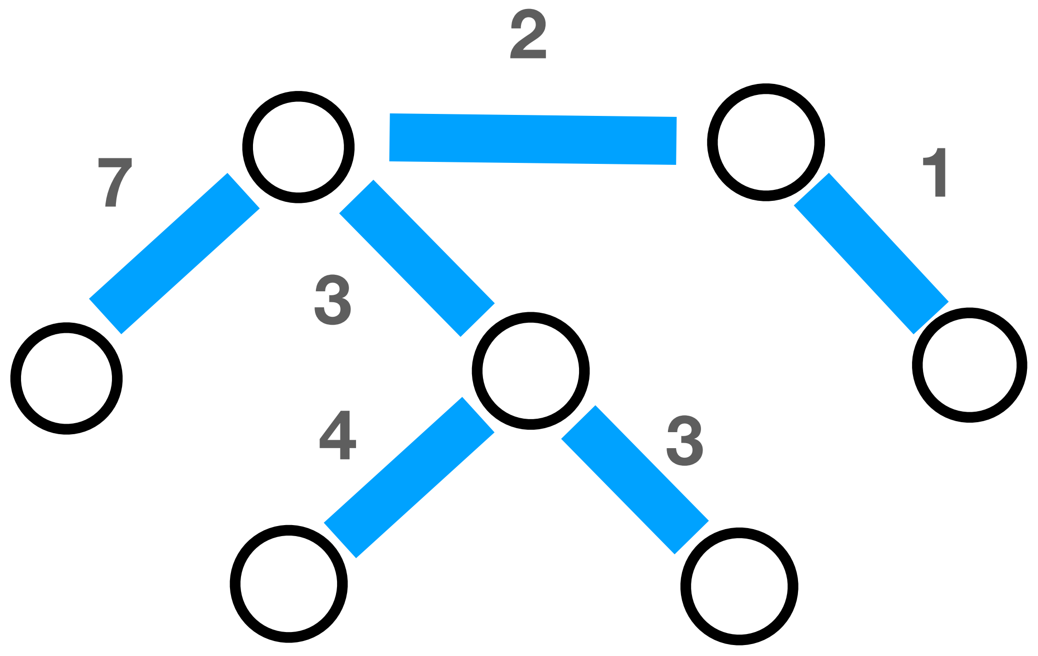
Fill Algorithm
- Flood Fill Algorithm

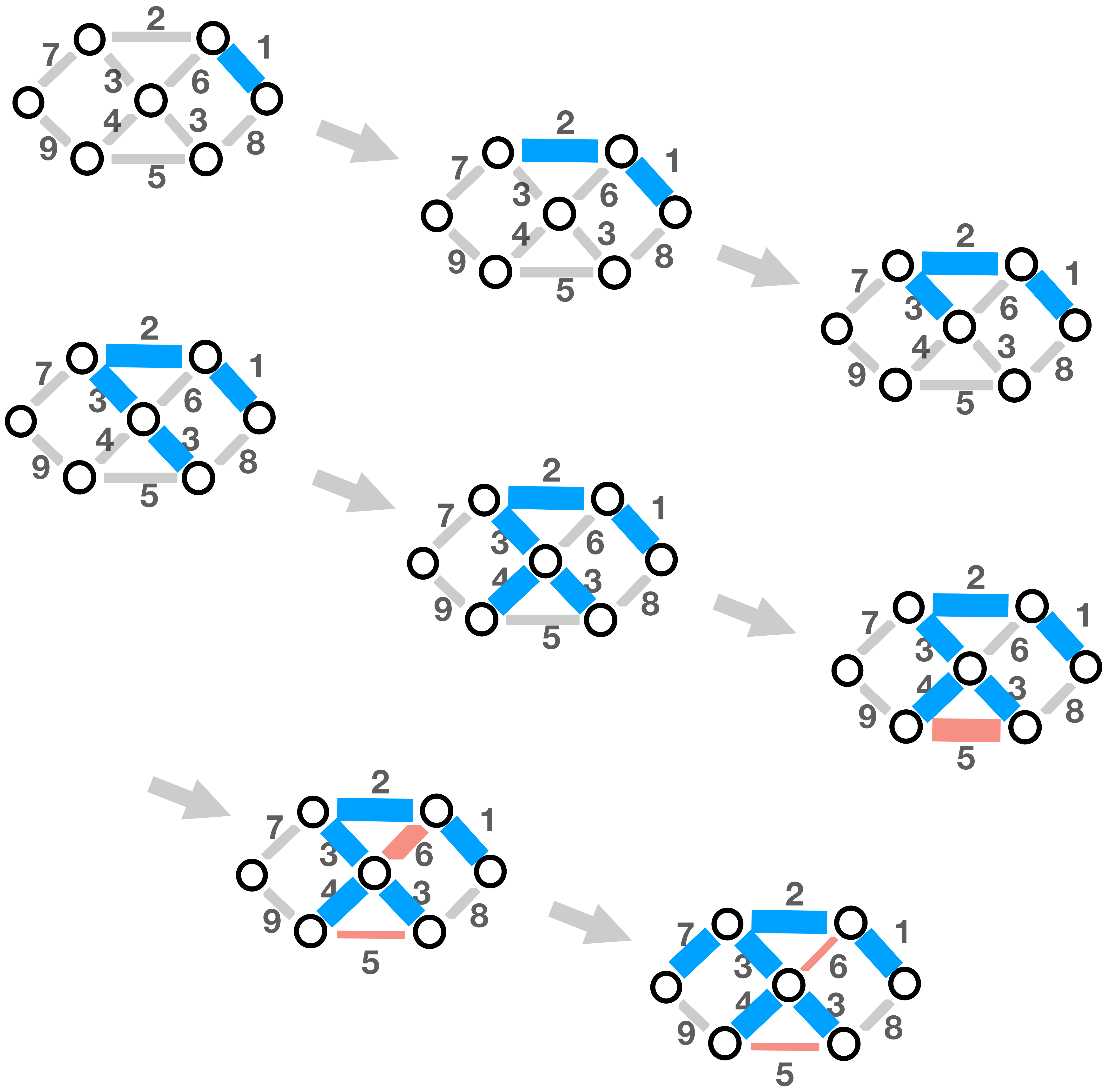
Kruskal's Algorithm:

…for minimum spanning tree

- **N vertices**
- min spanning tree has **n-1 edges**







Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Union-Find or Merge-Find

Computation Cost:
Cost: **O(|V|^3))**

Data Structure: Disjoint Sets

Two Operations:

- Find: find set that contains any element (represented by *representative element*)

- Union: combine sets

each set often stored as a tree…

Disjoint-set Forest

Set: represented as a tree

Representative: root of tree

Find: follows parent nodes to the root

Union: attach root of one set to root of the other

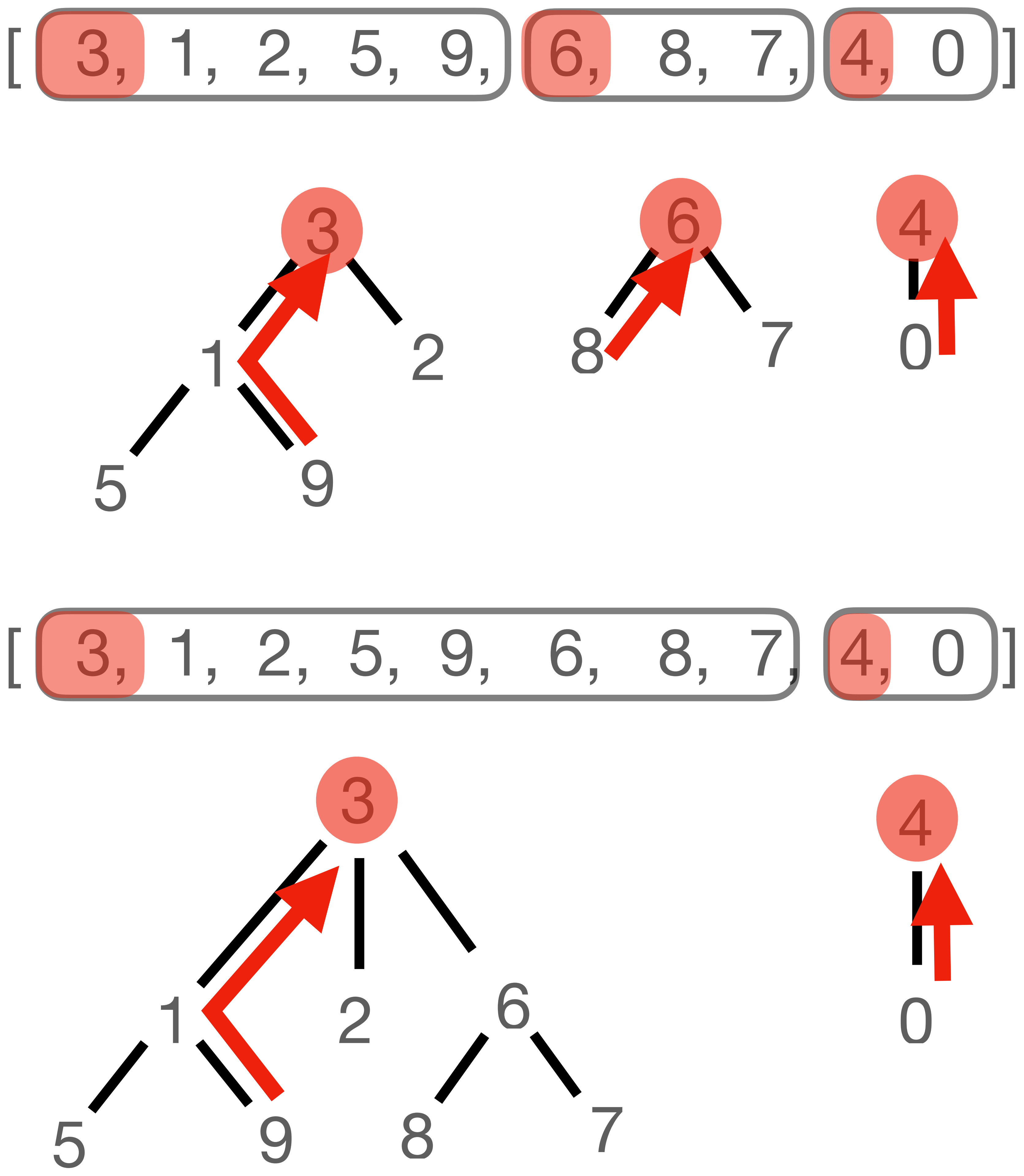
Not better than other ways to implement sets (ie. linked lists, etc) unless

1. Union by rank

Smaller (shallower) tree attached to root of larger (deeper) tree

2. Path compression

Move nodes up to too whenever Find is called



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Union-Find or Merge-Find

Computation Cost:
Cost: **O(|V|^3))**

Data Structure: Disjoint Sets

Two Operations:

- Find: find set that contains any element (represented by *representative element*)

- Union: combine sets

each set often stored as a tree…

Disjoint-set Forest

Set: represented as a tree

Representative: root of tree

Find: follows parent nodes to the root

Union: attach root of one set to root of the other

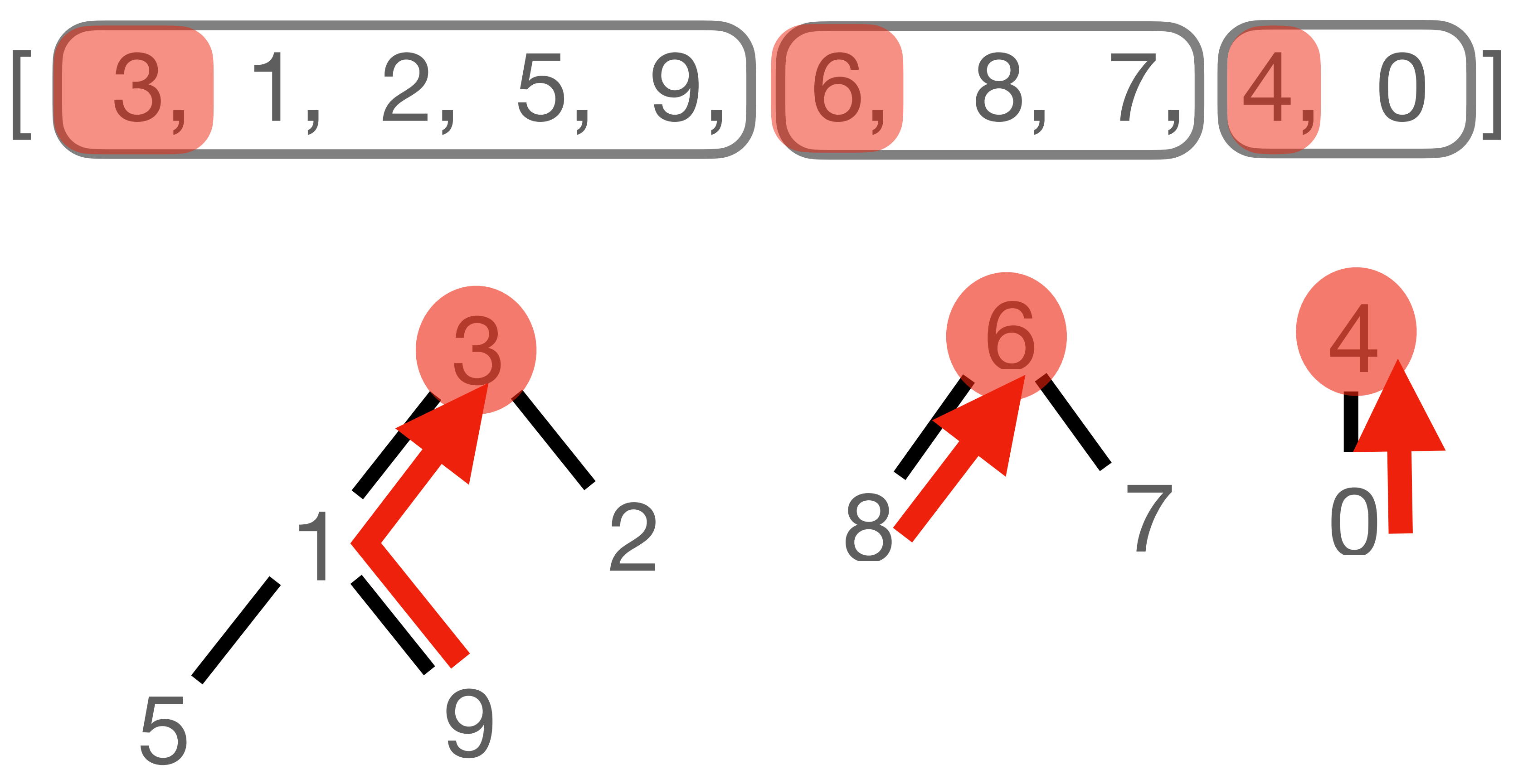
Not better than other ways to implement sets (ie. linked lists, etc) unless

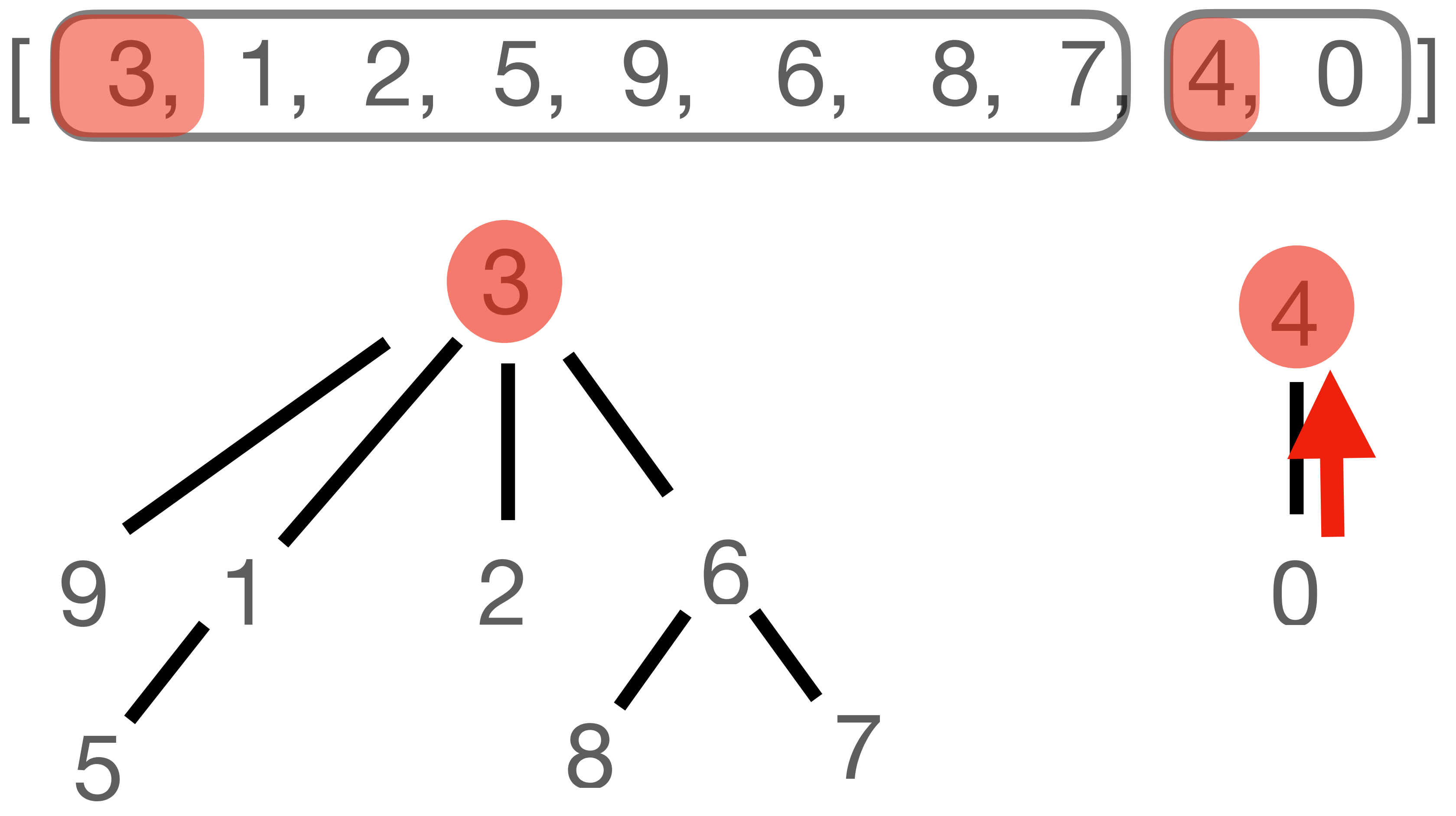
1. Union by rank

Smaller (shallower) tree attached to root of larger (deeper) tree

2. Path compression

Move nodes up to too whenever Find is called





Algorithms

Search Algorithms

- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms

- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting

- Kahn's topological sort (DAG)

Tree/Cycle algorithms

- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path

- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms

- Huffman Coding

Fill Algorithm

- Flood Fill Algorithm

Union-Find or Merge-Find

Computation Cost:
Cost: **O(|V|^3))**

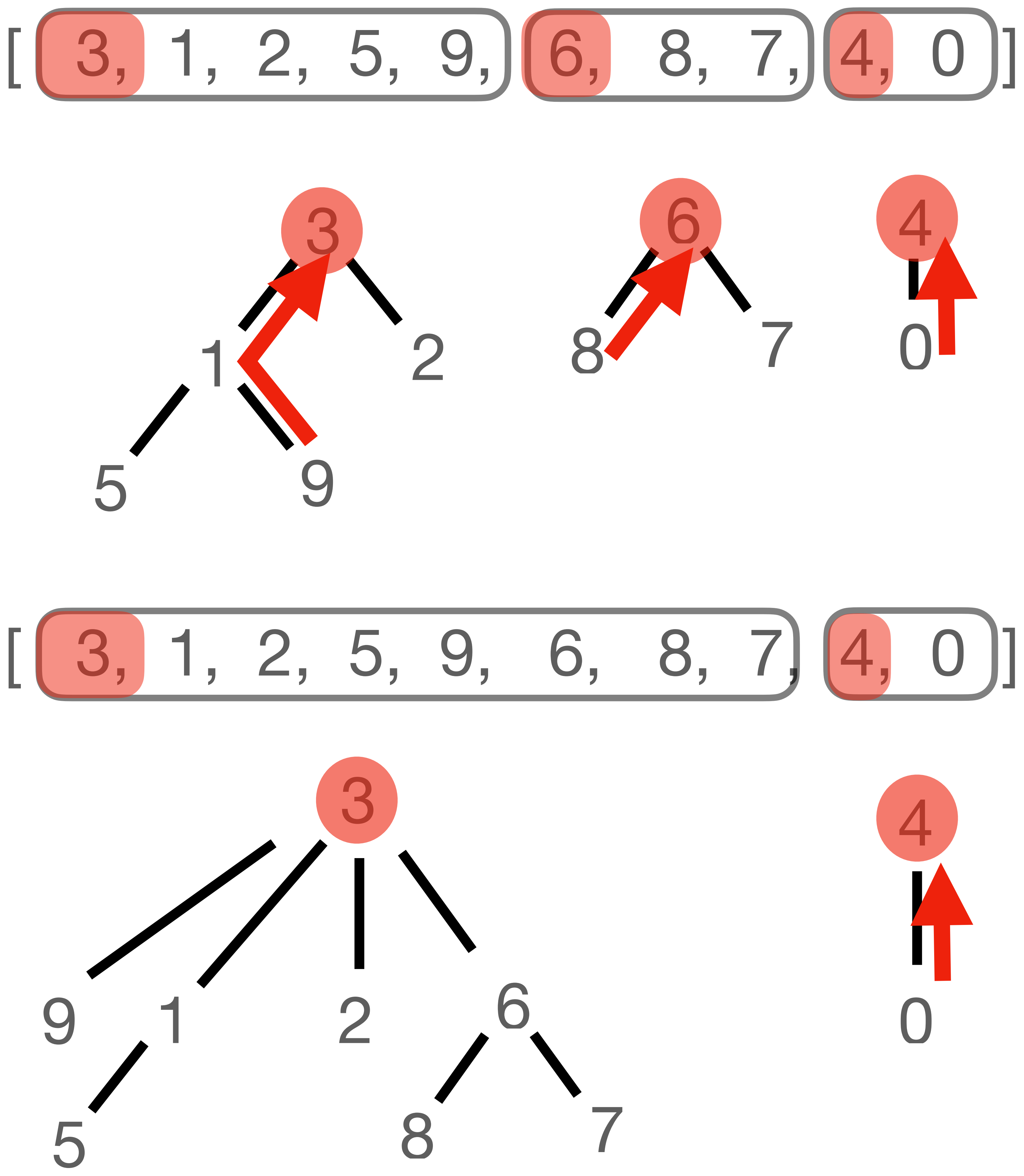
Data Structure: Disjoint Sets

Two Operations:

- **Find:** find set that contains any element
  set represented by *representative element*

- **Union:** combine sets

Disjoint Set Forest



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Union-Find Cycle Detection

Data Structure: Disjoint Sets

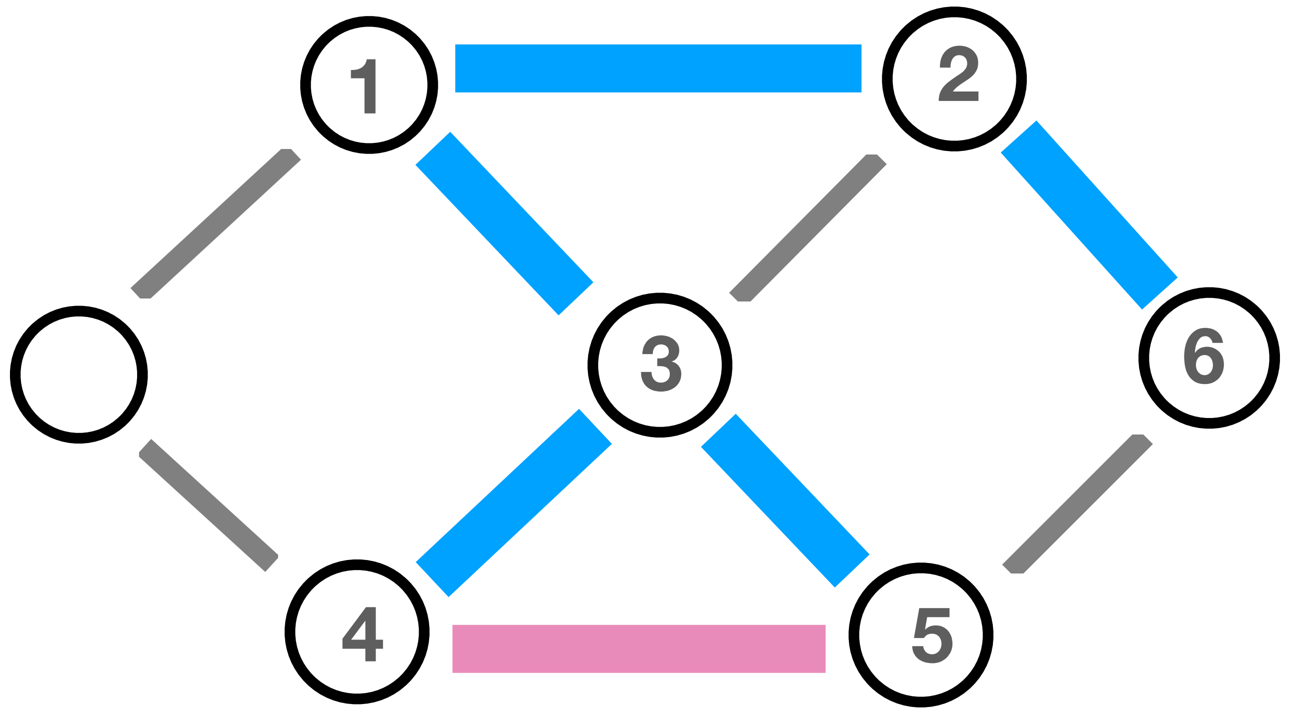
Two Operations:

- Find: find set that contains any element set by *representative element*

- Union: combine sets

Undirected Graph:

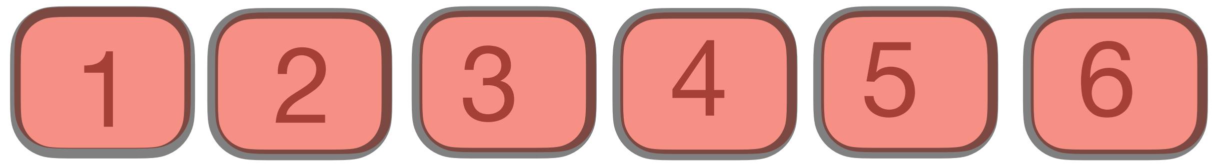
...minimum spanning tree step - cycle detection



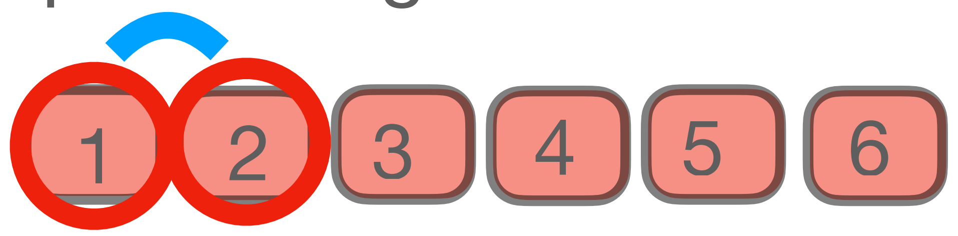
Cycle detected

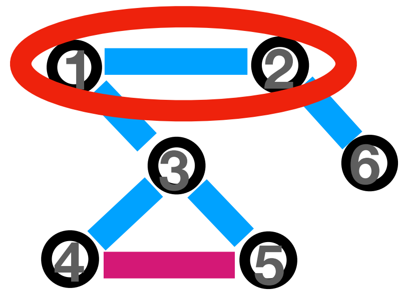
Disjoint Set Forest

disjoint sets: one with each vertex

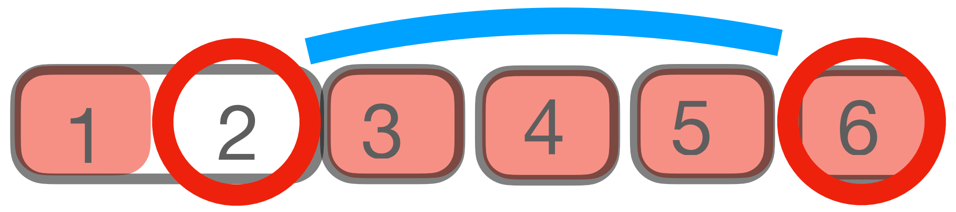


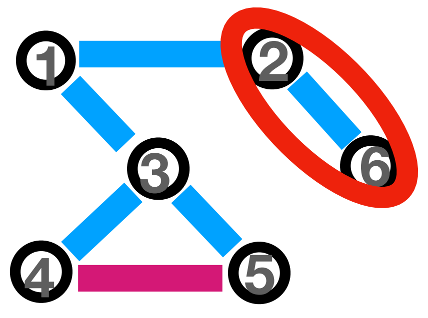
Loop over edges...



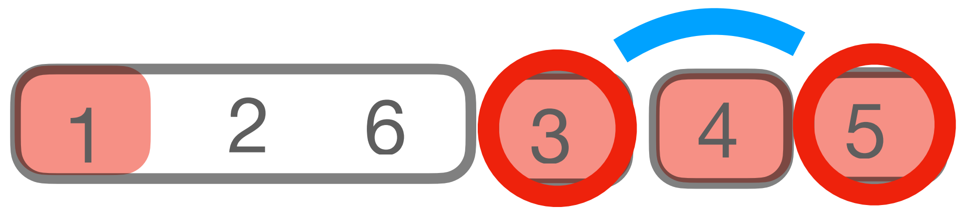


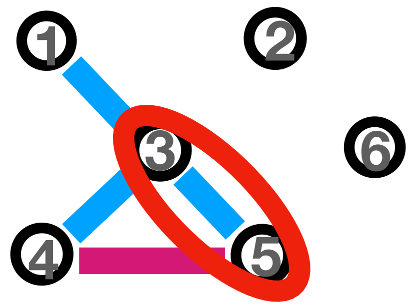
check if two vertices in the same set...
1 not = 2 ... so merge sets.



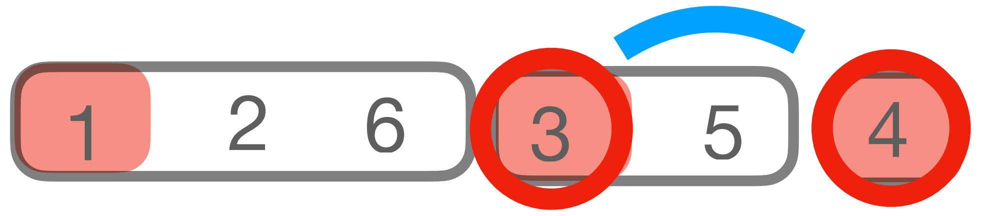


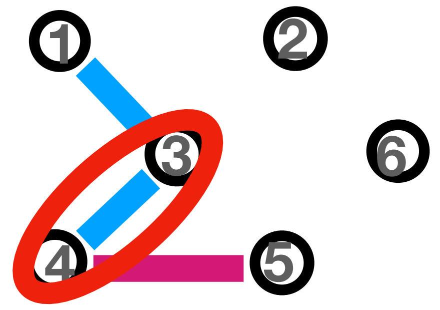
check if two vertices in the same set...
1 not = 6 ... so merge sets.



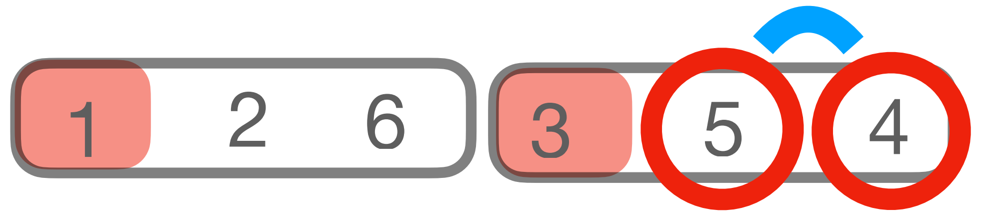


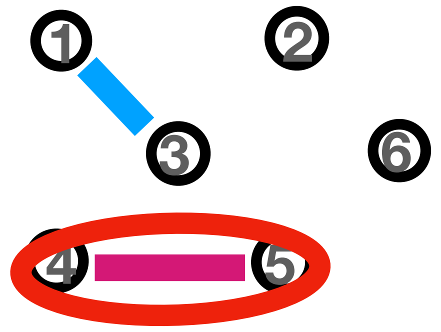
check if two vertices in the same set...
3 not = 5 ... so merge sets.





check if two vertices in the same set...
3 not = 5 ... so merge sets.





check if two vertices in the same set...
3 = 3 ... CYCLE DETECTED

Computation Cost:
Cost: **O(|V|^3))**

Algorithms

Dijkstra's Algorithm:

Unvisited

Visited

Computation Cost:
Cost: **O(|E| log(|V|))**

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

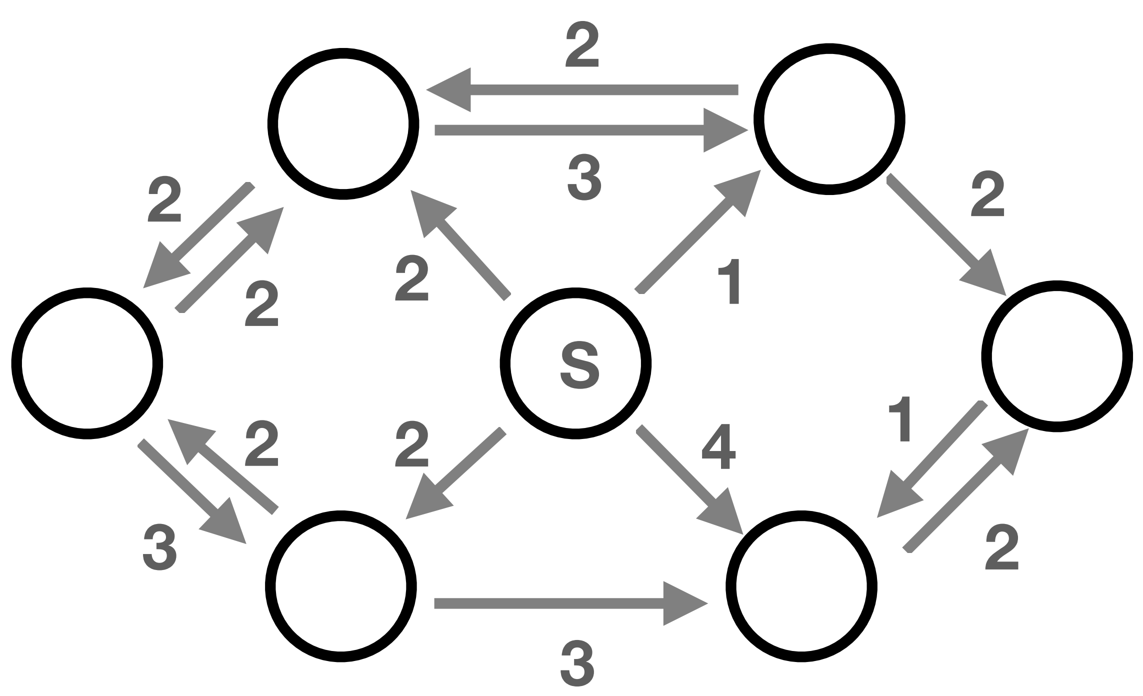
Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Shortest path from source on a weighted graph.
Edges have only positive weights.

Graph: G = (V,E)

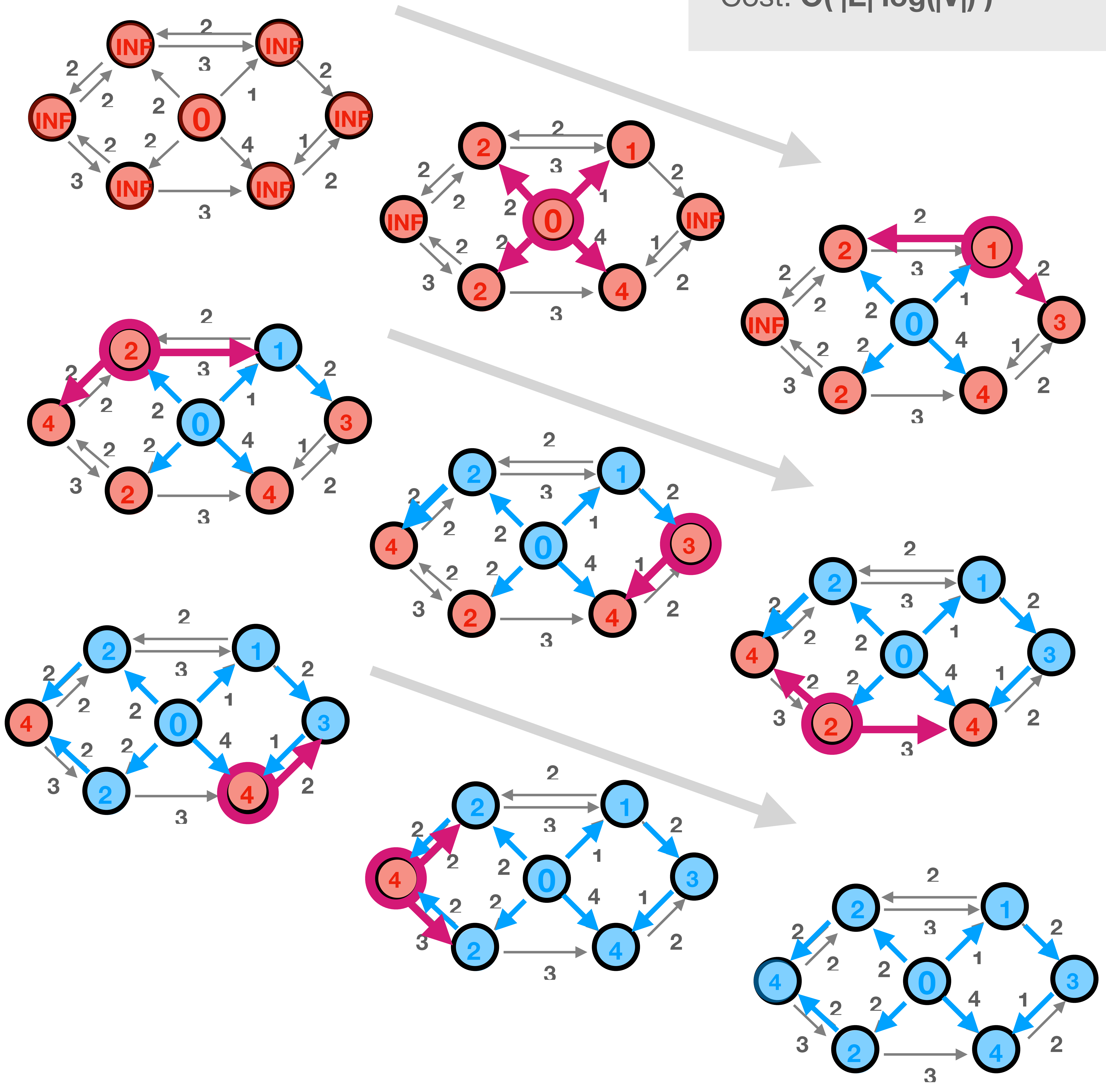


Idea/Pseudo-code:

Initialize:
Visited = {source}, cost = 0
Unvisited = { other nodes }, cost = INF

For each node:
Starting with lowest cost node…
1. update minimum cost to neighbors
2. Mark node as visited

Stop:
When each node has been visited or when desired node is reached.



Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms
Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

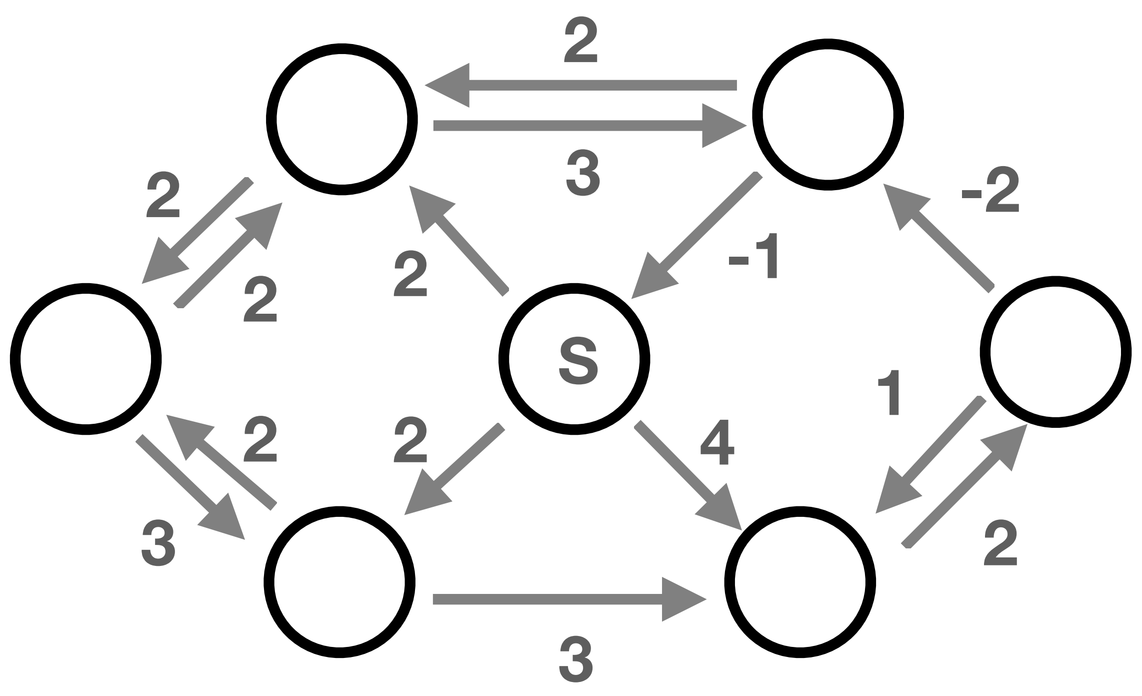
Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

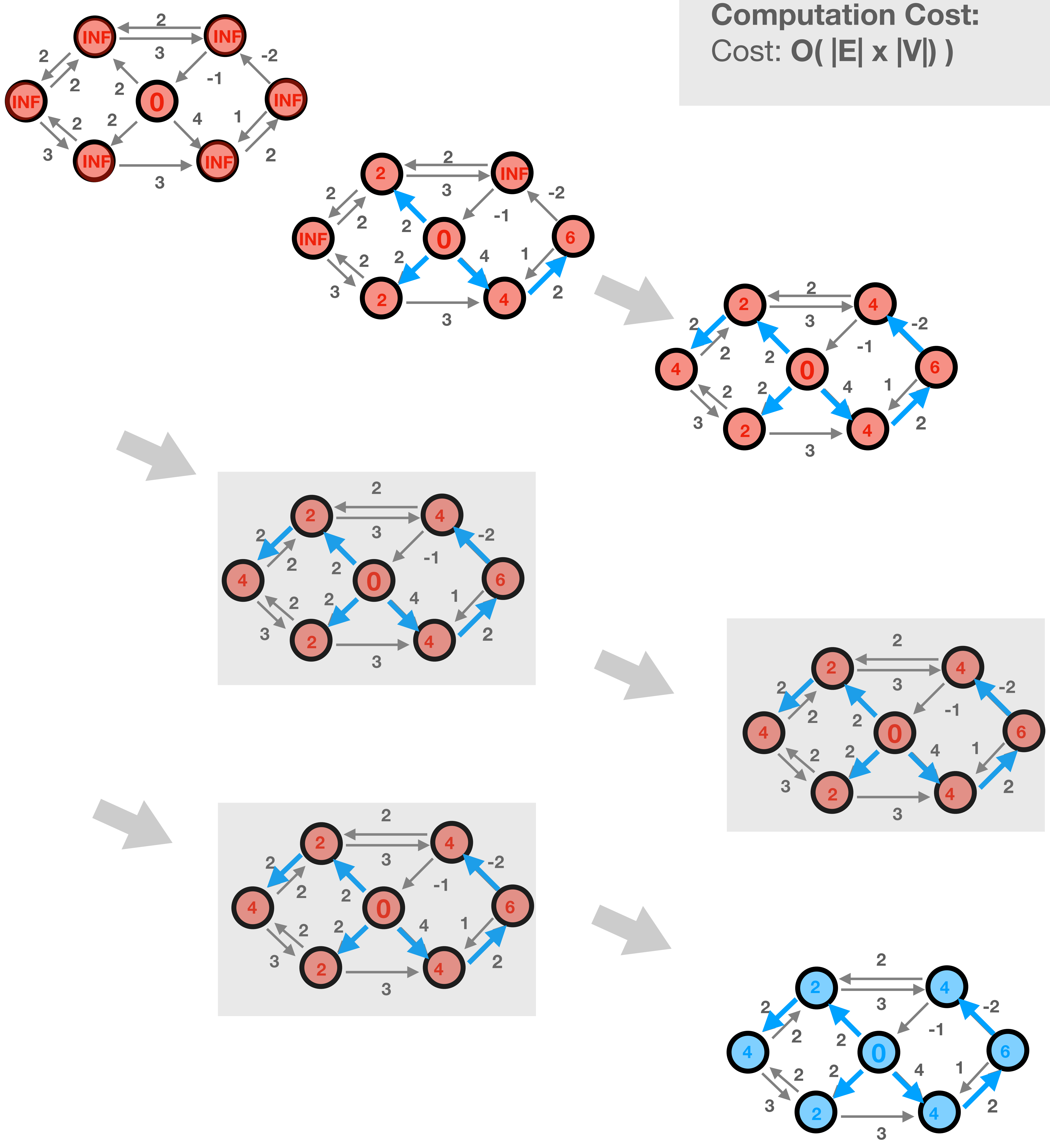
Bellman Ford Algorithm:

Shortest path from source on a weighted graph.
Edges have positive and negative weights
Report negative weight cycles.

Graph: G = (V,E)



Computation Cost:
Cost: **O(|E| x |V|))**



Idea/Pseudo-code:

Initialize:
Each node: cost = INF, parent = NULL

Source node: cost = 0
For |V|-1 iterations:
For each edge e:
Edge e: from u to v
If cost(v) > cost(u) + cost(edge):
cost(v) = cost(u) + cost(edge)
parent(v) = u

Else if cost(v) < cost(u)+cost(edge):
Report negative cycle

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

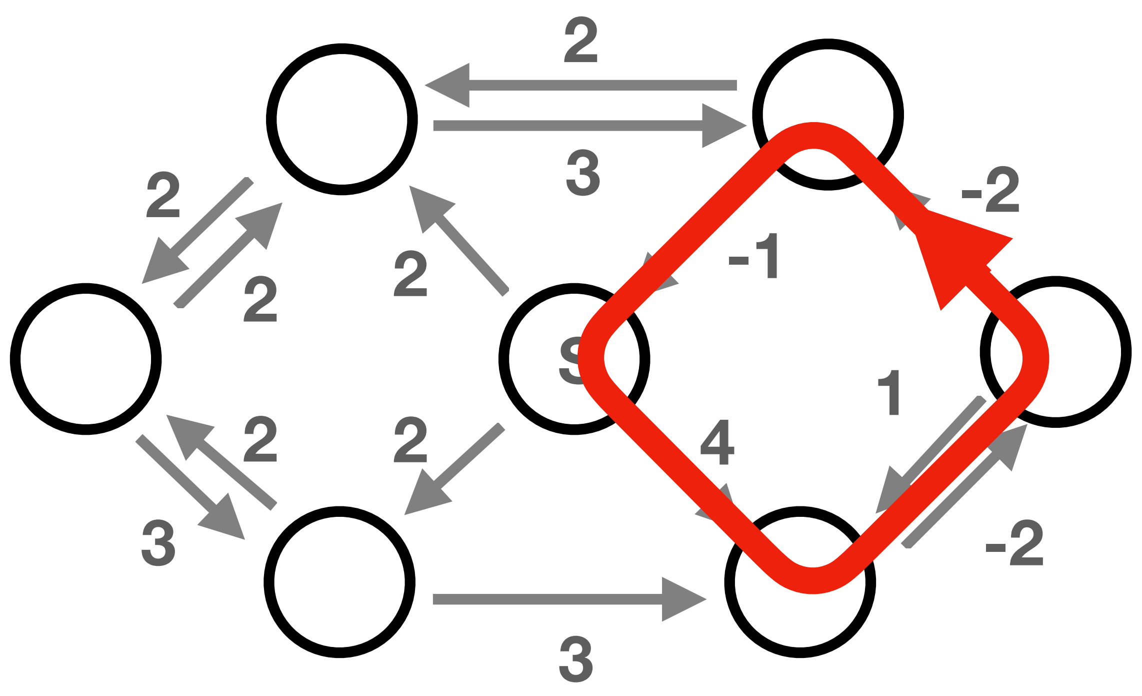
Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

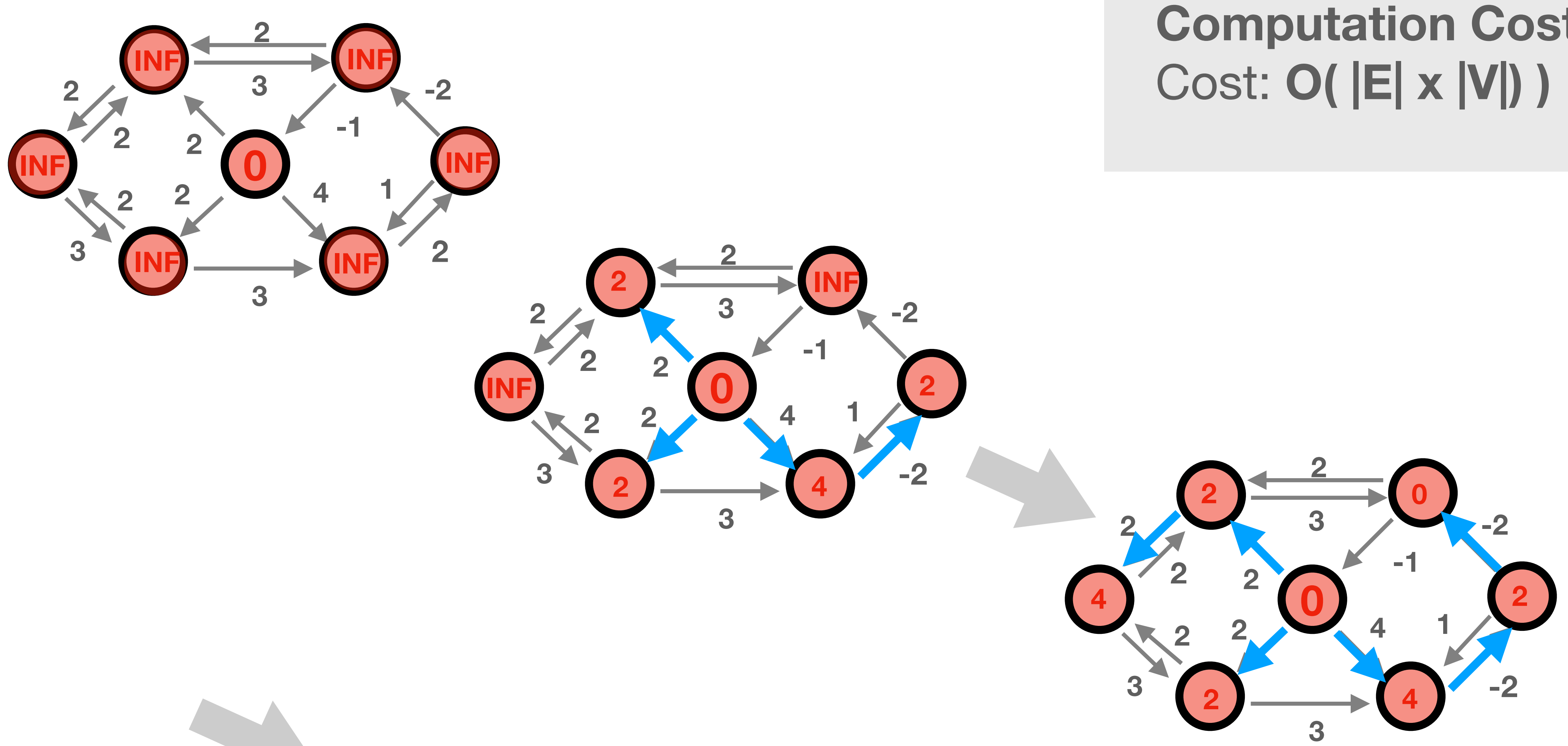
Bellman Ford Algorithm:

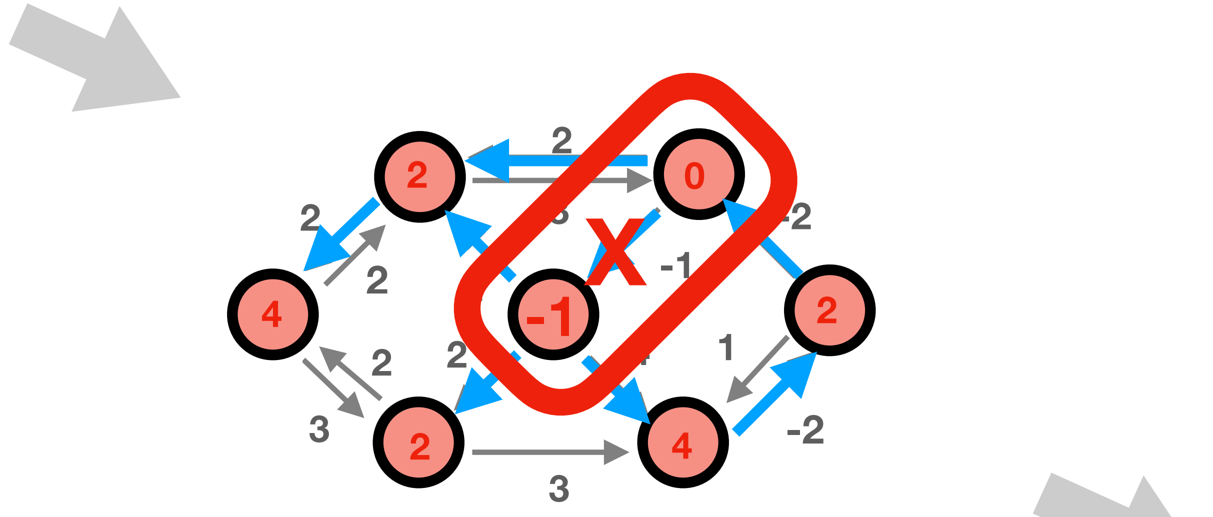
…with negative weight cycle…

Graph: G = (V,E)



… total weight is -1





Report cycle w/ negative weights

STOP

Computation Cost:
Cost: **O(|E| x |V|))**

Idea/Pseudo-code:

Initialize:
Each node: cost = INF, parent = NULL

Source node: cost = 0
For |V|-1 iterations:
For each edge e:
Edge e: from u to v
If cost(v) > cost(u) + cost(edge):
cost(v) = cost(u) + cost(edge)
parent(v) = u

Else if cost(v) < cost(u)+cost(edge):
Report negative cycle

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

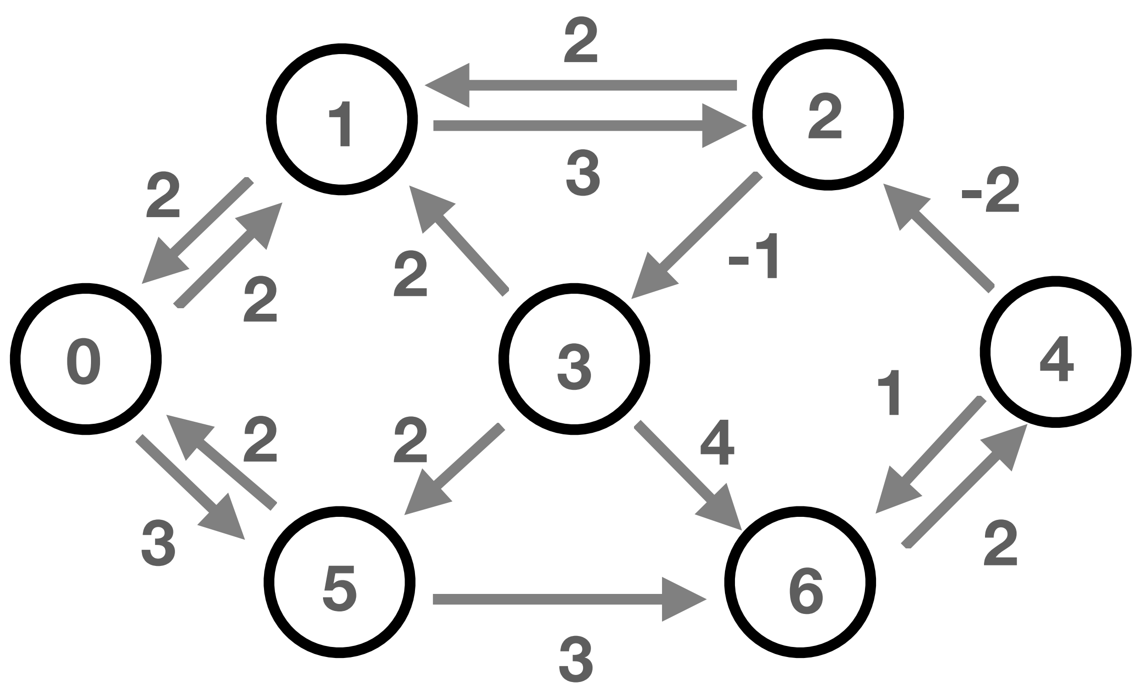
Fill Algorithm
- Flood Fill Algorithm

Floyd - Warshall algorithm:

Computation Cost:
Cost: **O(|V|^3))**

Shortest path from every node…
…to every other node

Graph: G = (V,E)



dist =

	0	1	2	3	4	5	6
0	0	INF	INF	INF	INF	INF	INF
1	INF	0	INF	INF	INF	INF	INF
2	INF	INF	0	INF	INF	INF	INF
3	INF	INF	INF	0	INF	INF	INF
4	INF	INF	INF	INF	0	INF	INF
5	INF	INF	INF	INF	INF	0	INF
6	INF	INF	INF	INF	INF	INF	0

dist =

	0	1	2	3	4	5	6
0	0	2	INF	INF	INF	3	INF
1	2	0	3	INF	INF	INF	INF
2	INF	2	0	-1	INF	INF	INF
3	INF	2	INF	0	INF	2	4
4	INF	INF	-2	INF	0	INF	1
5	2	INF	INF	INF	INF	0	3
6	INF	INF	INF	INF	2	INF	0

Idea/Pseudo-code:

```
Initialize:
    dist matrix: |V| x |V|
    Set diagonal elements = 0
    if edge from i to j in graph:
        dist[i][j] = weight;

For k = 0 to |V|-1:
    For j = 0 to |V|-1:
        For i = 0 to |V|-1:
        If dist[i][k] + dist[k][j] < dist[i][j]:
            dist[i][j] = dist[i][k] + dist[k][j]

After:
if dist[i][i] is negative for any I:
    Report negative cycle
```

Algorithms

Floyd - Warshall algorithm:

Computation Cost:
Cost: **O(|V|^3))**

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

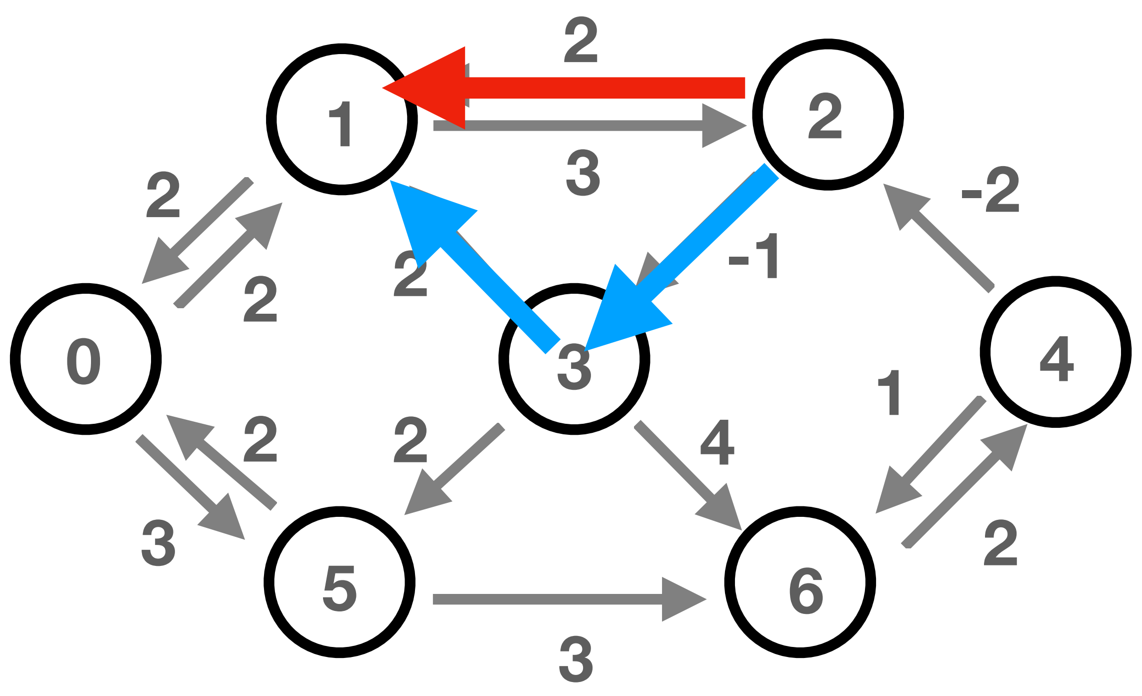
Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Shortest path from every node…
…to every other node

Graph: G = (V,E)



Idea/Pseudo-code:

Initialize:
dist matrix: |V| x |V|
Set diagonal elements = 0
if edge from i to j in graph:
dist[i][j] = weight;

For k = 0 to |V|-1:
For j = 0 to |V|-1:
For i = 0 to |V|-1:
If dist[i][k] + dist[k][j] < dist[i][j]:
dist[i][j] = dist[i][k] + dist[k][j]

After:
if dist[i][i] is negative for any I:
Report negative cycle

dist =

	0	1	2	3	4	5	6
0	0	INF	INF	INF	INF	INF	INF
1	INF	0	INF	INF	INF	INF	INF
2	INF	INF	0	INF	INF	INF	INF
3	INF	INF	INF	0	INF	INF	INF
4	INF	INF	INF	INF	0	INF	INF
5	INF	INF	INF	INF	INF	0	INF
6	INF	INF	INF	INF	INF	INF	0

dist =

	0	1	2	3	4	5	6
0	0	2	INF	INF	INF	3	INF
1	2	0	3	INF	INF	INF	INF
2	INF	2	0	-1	INF	INF	INF
3	INF	2	INF	0	INF	2	4
4	INF	INF	-2	INF	0	INF	1
5	2	INF	INF	INF	INF	0	3
6	INF	INF	INF	INF	2	INF	0

dist =

	0	1	2	3	4	5	6
0	0	2	INF	INF	INF	3	INF
1	2	0	3	INF	INF	INF	INF
2	INF	2	0	-1	INF	INF	INF
3	INF	2	INF	0	INF	2	4
4	INF	INF	-2	INF	0	INF	1
5	2	INF	INF	INF	INF	0	3
6	INF	INF	INF	INF	2	INF	0

If 2 > -1 + 2 :

dist[2,1] = -1 + 2 = 1

dist =

	0	1	2	3	4	5	6
0	0	2	INF	INF	INF	3	INF
1	2	0	3	INF	INF	INF	INF
2	INF	1	0	-1	INF	INF	INF
3	INF	2	INF	0	INF	2	4
4	INF	INF	-2	INF	0	INF	1
5	2	INF	INF	INF	INF	0	3
6	INF	INF	INF	INF	2	INF	0

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms

Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Huffman Coding - Data Compression

Computation Cost:
Cost: **O(nlog(n))**

Fixed length encoding: "every character takes 8 bits."

Variable length encoding: "Use less memory for more frequency characters."

Character	A	B	C	D	E
Frequency	11	3	7	5	20
Binary Encoding	1	100	10	11	0

Problem: "Encoding is ambiguous"

Ex. 010010111 → EAEEAEAAA, EBCDA, EAEECAD, …etc

Solution: "No code (for a character) can be a prefix of another code

Character	E	A	C	D	B
Frequency	20	11	7	5	3
Binary Encoding	0	10	110	1110	1111

Algorithms

Search Algorithms
- Binary search
- Breadth First Search (BFS)
- Depth First Search (DFS)
- Kadane's algorithm
- KMP algorithm
- Quick select algorithm
- Boyer-Moore Majority Vote algorithm
- Euclid's algorithm

Sorting Algorithms
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort
- Introsort
- Bubble sort
- Non-comparison sorts
- Counting sorts
- Radix sort
- Bucket sort
- (Kahn's topological sort)

Graph Algorithms
Sorting
- Kahn's topological sort (DAG)

Tree/Cycle algorithms
- Floyd's Cycle Detection Algorithm
- Union-Find Cycle Detection
- Kruskal's Algorithm

Shortest path
- Dijkstra's Algorithm
- Bellman Ford Algorithm
- Floyd Warshall Algorithm
- Lee Algorithm

Compression Algorithms
- Huffman Coding

Fill Algorithm
- Flood Fill Algorithm

Huffman Coding - Data Compression

Computation Cost:
Cost: **O(nlog(n))**

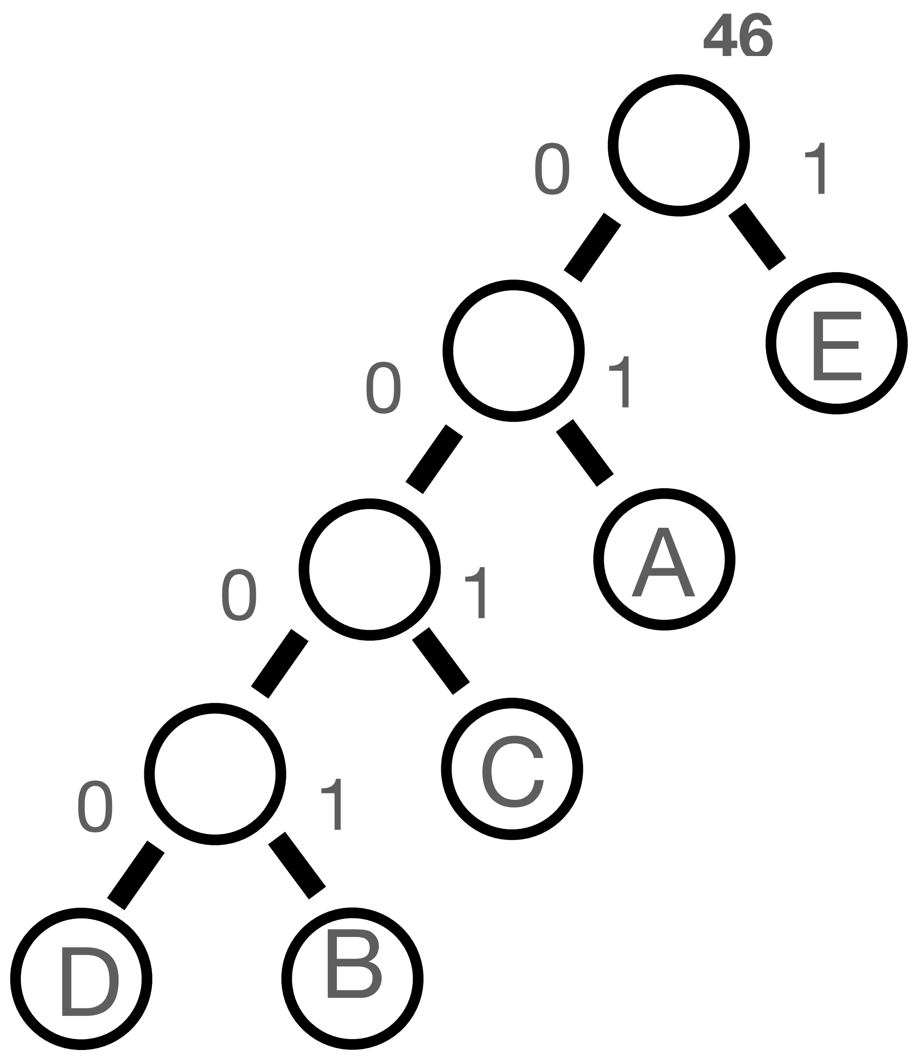
Huffman coding: Create binary tree

Required data structure: Priority queue

Character	E	A	C	D	B
Frequency	20	11	7	5	3
Binary Encoding	1	01	001	0001	0000

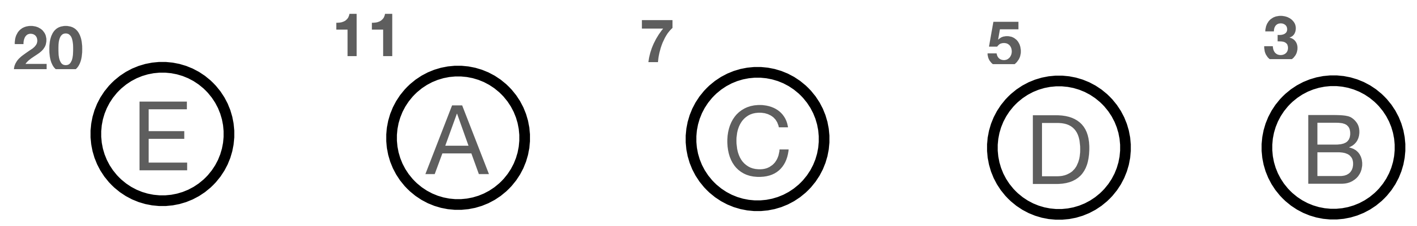
Binary Tree:

weights along branches give encoding

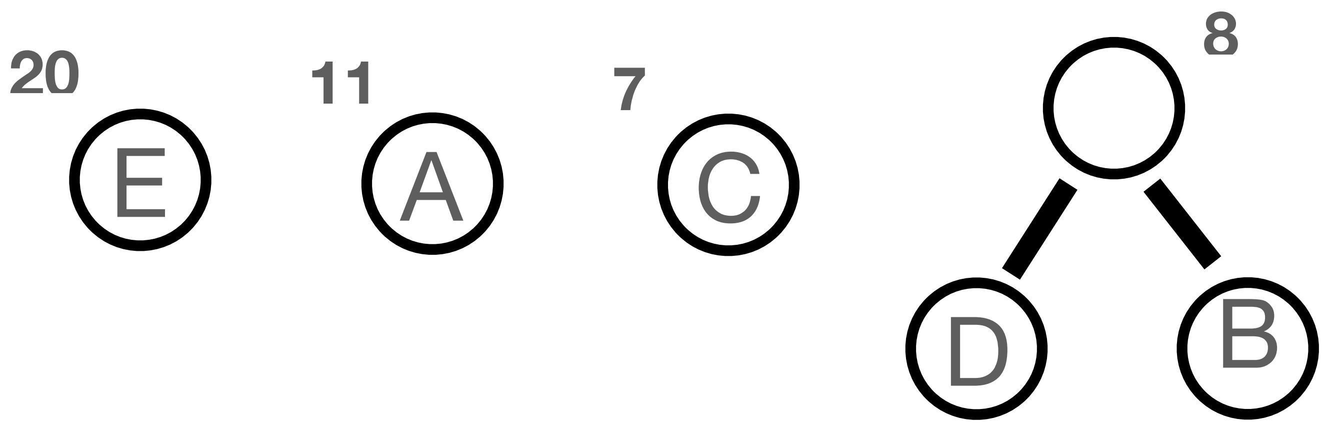


Note the ordering: …binary tree

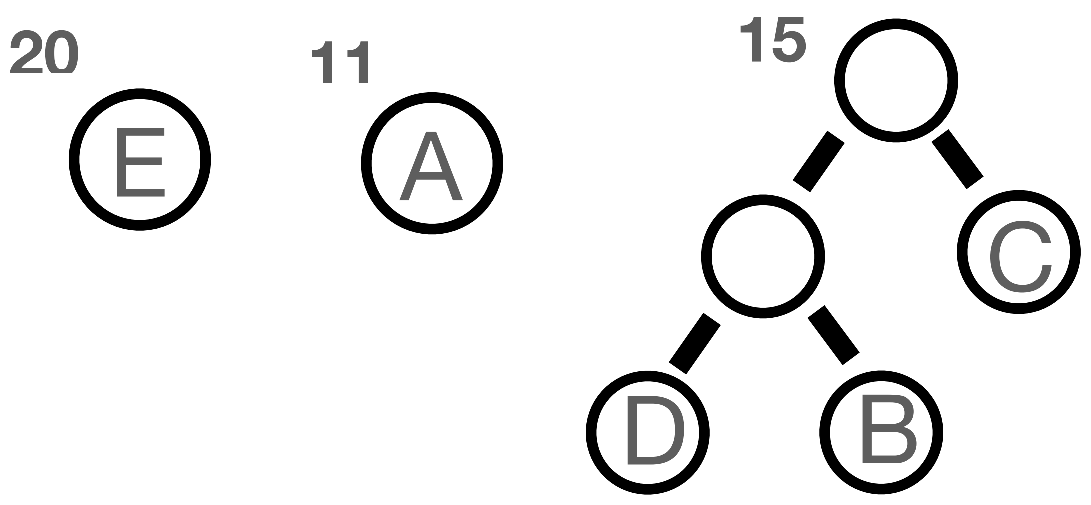
… more frequent characters (or groups of characters) assigned 0



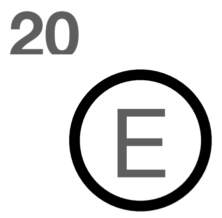
Pick two elements with lowest frequency…

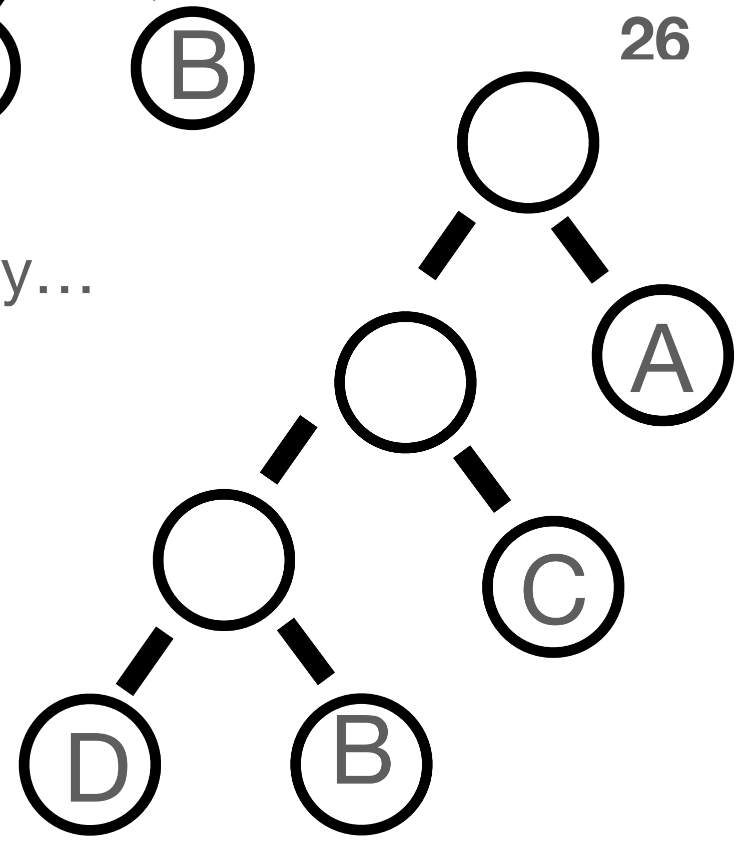


Pick two elements with lowest frequency…



Pick two elements with lowest frequency…





and so on…