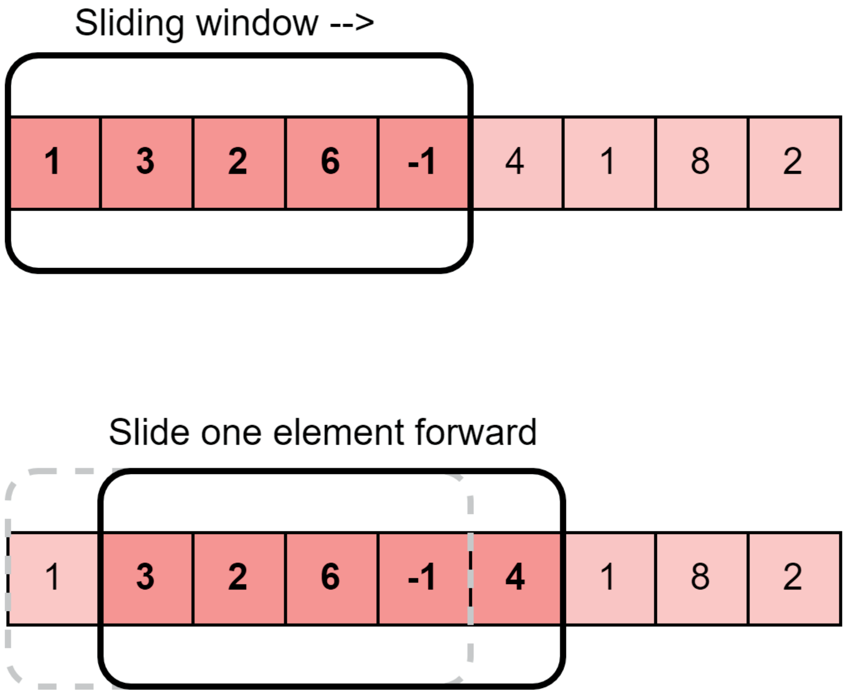
12 Must Know Coding Patterns & Their Applications in System Design Interview

DesignGurus.io

1. Sliding Window

Maintains a "window" of elements in a data structure (usually an array or a string) to optimize the solution of a problem.



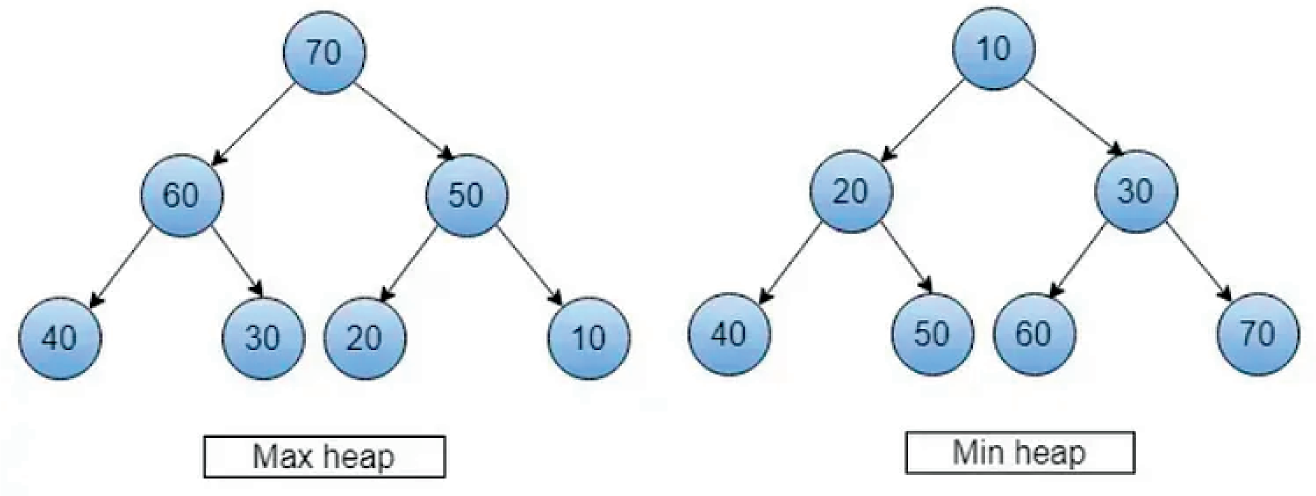
Usage:

- Network congestion control algorithms (like TCP)
- Data compression algorithms (like LZ77)

DesignGurus.io

2. Two Heaps

Uses two heaps (min-heap and max-heap) to maintain a specific order of elements to efficiently solve problems



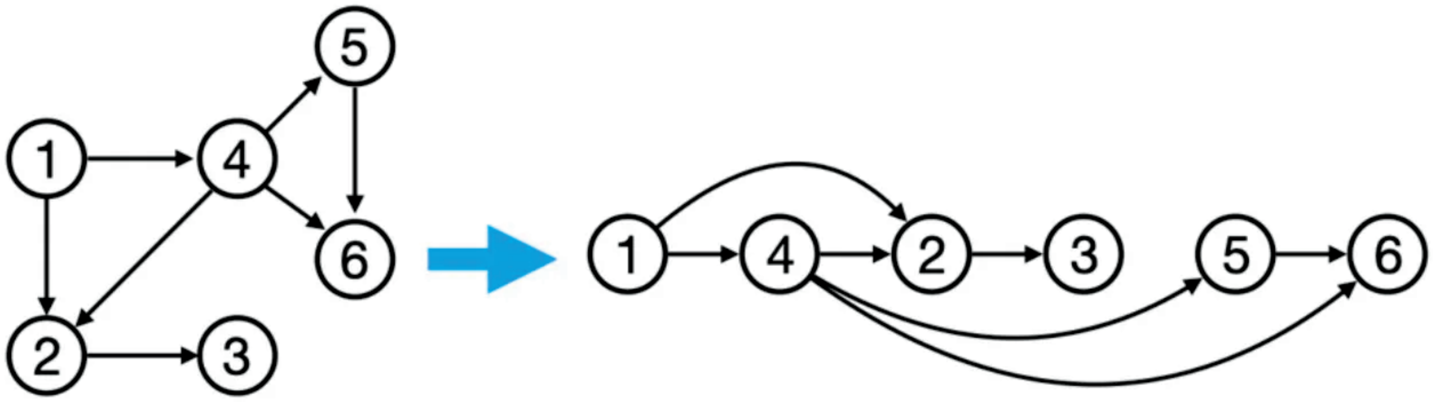
Usage:

- Managing a priority queue in a schedule.
- Implementing Dijkstra's shortest path algorithm.
- Maintaining the median of a dynamic data set.

DesignGurus.io

3. Topological Sort

Used for linear ordering of the vertices of a directed acyclic graph (DAG) such that for every directed edge (u, v), vertex u comes before vertex v in the ordering.



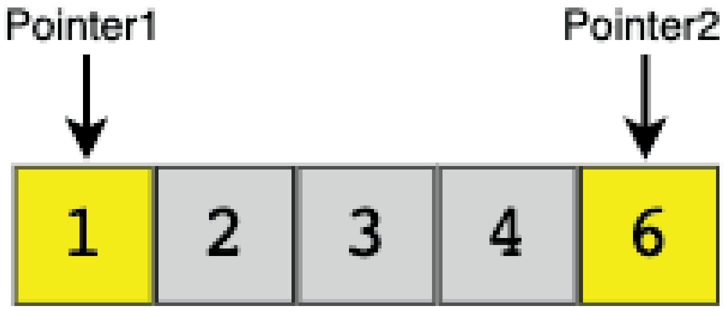
Usage:

- Scheduling tasks with dependencies.
- Determining the order of compilation for a set of source files.

DesignGurus.io

4. Two Pointers

Uses two pointers that move through the data structure (often an array) in a coordinated manner to solve problems.



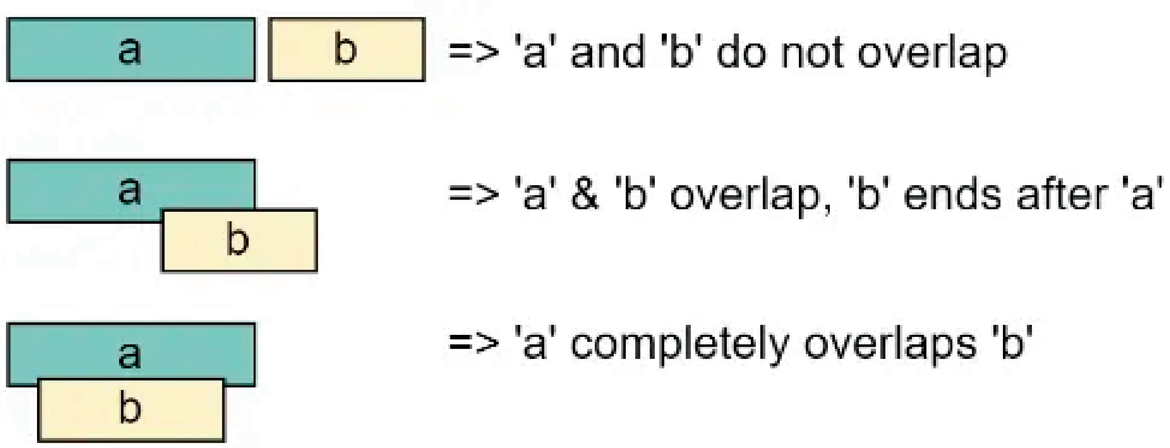
Usage:

- Merge-sort algorithm.
- Binary search.

DesignGurus.io

5. Merge Intervals

Involves merging of overlapping or continuous intervals in a given data structure (usually an array or a list) to optimize solutions for a specific problem.


a
b
=> 'a' and 'b' do not overlap
a
b
=> 'a' & 'b' overlap, 'b' ends after 'a'
a
b
=> 'a' completely overlaps 'b'

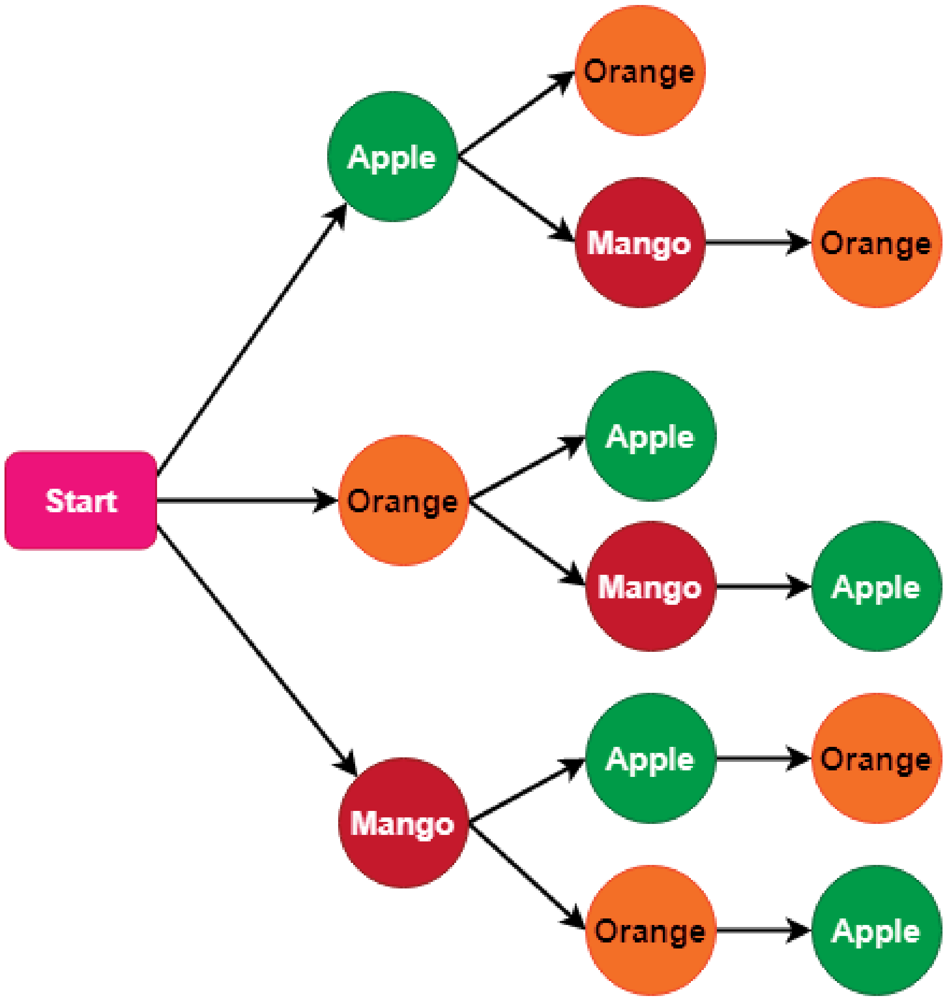
Usage:

- Scheduling meeting rooms.
- Managing calendar events.
- Optimizing resource allocation.

DesignGurus.io

6. Backtracking

Tries out different possibilities, undoing them, and then trying out new paths until a solution is found.



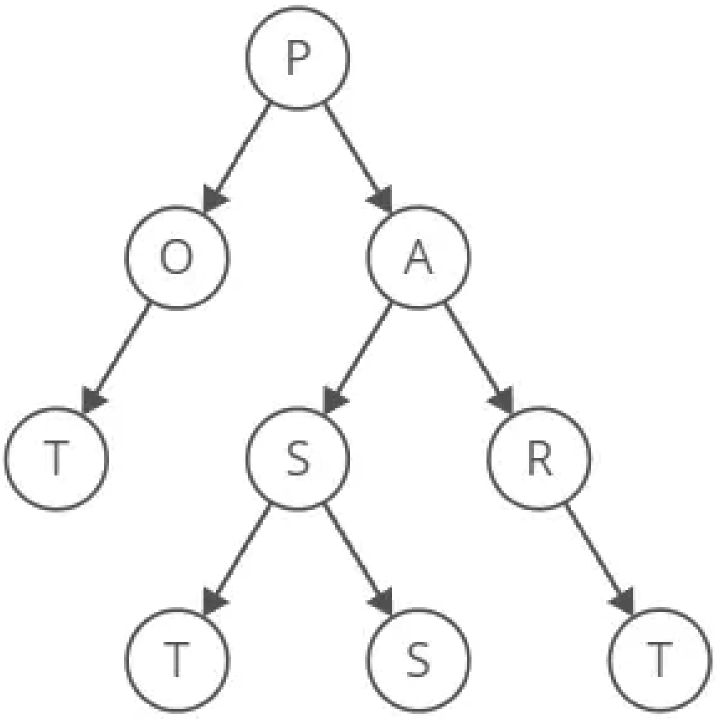
Usage:

- Solving Sudoku puzzles.
- Generating permutations and combinations.

DesignGurus.io

7. Trie (Prefix Tree)

Uses a tree-like data structure to efficiently store and retrieve strings based on their prefixes.



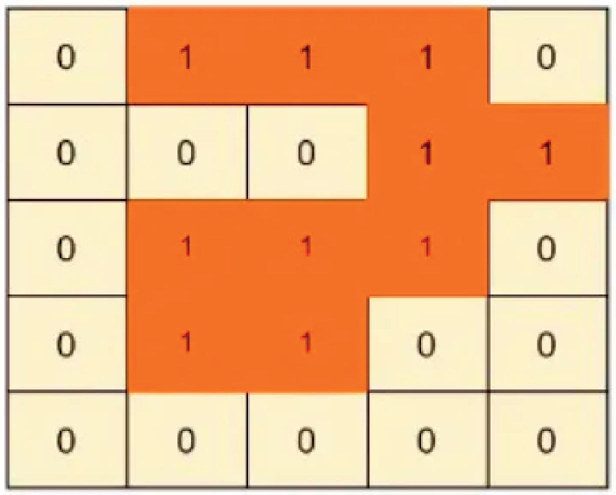
Usage:

- Implementing an autocomplete system.
- Spell checkers.
- IP routing (Longest Prefix Matching).

DesignGurus.io

8. Flood Fill

Traverses a 2D grid (matrix) and replacing connected elements of a specific value with a new value.



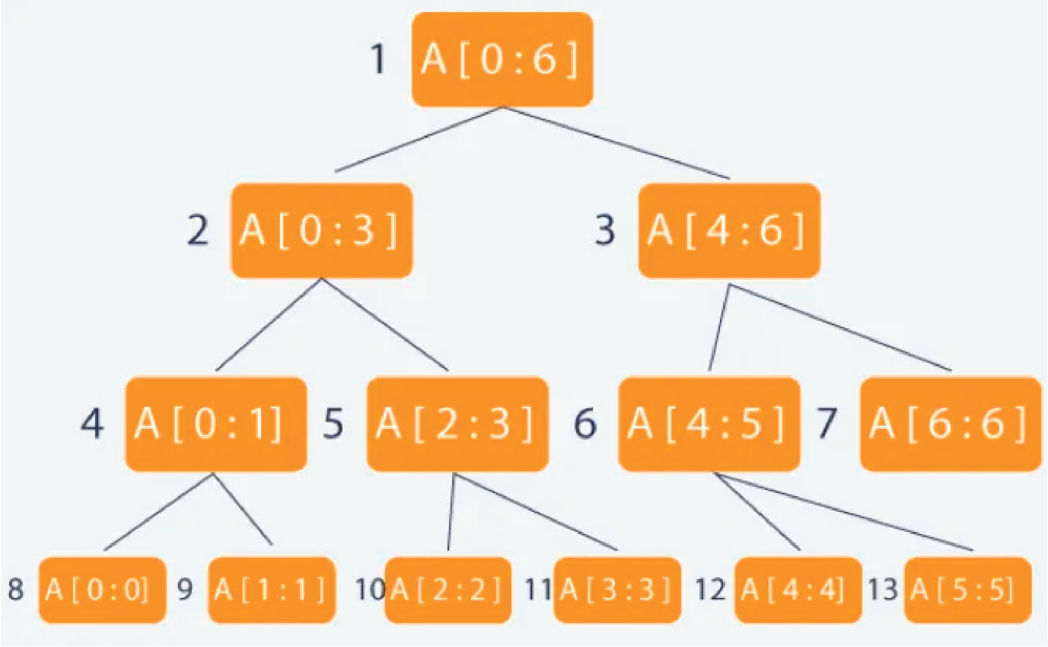
0	1	1	1	0
0	0	0	1	1
0	1	1	1	0
0	1	1	0	0
0	0	0	0	0

Usage:

- Filling a bounded area in graphics editors (like MS Paint).
- Counting connected regions in a 2D grid.

9. Segment Tree

Uses a tree-like data structure to efficiently perform range queries and updates on an array or a list.



Usage:

- Range queries in databases.
- Calculating range-based statistics (e.g., minimum, maximum, sum).

DesignGurus.io

10. Breadth-First Search (BFS)

Traverses a tree or a graph using a breadth-first approach, visiting all nodes at the current level before moving to the next level.

Usage:

- Web crawlers.
- Social network analysis (finding friends of friends) Routing algorithms (like OSPF) in networking.

DesignGurus.io

11. Depth-First Search (DFS)

Traverses a tree or a graph using a depth-first approach, visiting a node and exploring as far as possible along a branch before backtracking.

Usage:

- Solving mazes.
- Finding connected components in a graph.
- Generating permutations or combinations.

12. Union-Find (Disjoint Set)

Uses a data structure to keep track of disjoint sets and efficiently perform union and find operations on them.

Usage:

- Network connectivity
- Finding connected components in a graph

➡ Follow these techniques to distinguish yourself from others!

➡ Learn more about these patterns in "Grokking the Coding Interview" and "Grokking the System Design Interview" from **DesignGurus.io**