Algorithms — Container Design & Algorithm Selection

Answer all questions.
You must justify your design choices.
Big-O without explanation earns limited credit.
Diagrams are encouraged if they clarify reasoning.

Question 1 — Priority Matters (25 points)

You are designing a system that processes jobs with the following properties:

- Each job has:
  - an ID
  - a priority (integer)
  - a timestamp
- The system must:
  - Insert new jobs frequently
  - Always process the highest priority job next
  - Occasionally change the priority of an existing job

(a) (5 points) What abstract data structure best models this problem?

(b) (5 points) Choose a concrete container implementation.

(c) (5 points) Choose the primary algorithm(s) used for insertion and removal.

(d) (5 points) State the time complexity for:

- insertion
- removal

(e) (5 points) Explain why a singly linked list with sorted insertion is inferior.

Question 2 — Undo, But Make It Efficient (20 points)

You are implementing an undo system for a text editor.

Requirements:

- Undo operations must occur in reverse order
- Undo must be fast
- Redo support is not required (for now)

(a) (5 points) What data structure is most appropriate?

(b) (5 points) Describe the invariant of this structure.

(c) (5 points) State the time complexity of undo.

(d) (5 points) Explain why a queue would be a poor choice.

Question 3 — Fast Search, Rare Updates (25 points)

You are given a dataset of 1 million integers.

Constraints:

- The data is loaded once at startup
- Searches are extremely frequent
- Insertions and deletions are rare
- Memory locality matters

(a) (5 points) Choose a container.

(b) (5 points) Choose a search algorithm.

(c) (5 points) State the time complexity of search.

(d) (5 points) Explain why this design favors cache performance.

(e) (5 points) Explain why a linked list is unacceptable.

Question 4 — Traversal With Intent (15 points)

You are working with a binary search tree.

Tasks:

- Output all values in sorted order
- Delete the entire tree safely from memory

(a) (5 points) Which traversal produces sorted output?

(b) (5 points) Which traversal should be used to delete the tree?

(c) (5 points) Explain why using the wrong traversal is dangerous.

Question 5 — Graph Reality Check (15 points)

You are given a sparse graph with:

- Millions of vertices
- Very few edges per vertex

Tasks:

- Traverse the entire graph
- Detect cycles

(a) (5 points) Choose a graph representation.

(b) (5 points) Choose a traversal algorithm.

(c) (5 points) State the time complexity in terms of V and E.

Question 6 — Design Judgment (Bonus 10 points)

You may answer **one** of the following:

- Describe a situation where a worse asymptotic complexity is acceptable.
- Explain why Big-O alone is insufficient to choose a data structure.

Clarity > length.