

Lecture 11-heaps

Question 1:

What is the worst-case time complexity of `removeMin()` in an array-based binary heap?

- A. $O(1)$
- B. $O(n)$
- C. $O(\log n)$
- D. $O(n \log n)$

Answer:

Question 2:

In a min-heap, every node must satisfy which condition?

- A. It is greater than its parent
- B. It is smaller than or equal to its children
- C. It is greater than its children
- D. It is smaller than its parent

Answer:

Question 3:

In the array representation of a heap, what is the index of the left child of a node at index i ?

- A. $2i$
- B. $2i+1$
- C. $i/2$
- D. $2i+2$

Answer:

Question 4:

Which step is not part of the `removeMin()` operation in a binary heap?

- A. Replace the root with the last node
- B. Percolate the new root down
- C. Swap the root with the smallest child
- D. Sort the entire heap

Answer: D.

Question 5:

A complete binary tree:

- A. Must be a perfect binary tree
- B. Has all levels filled except the last, which is filled left to right

- C. Allows gaps in the last level
- D. Requires every node to have exactly two children

Answer:

Question 6:

What is the time complexity of building a heap using Floyd's buildHeap algorithm?

- A. $O(n \log n)$
- B. $O(n)$
- C. $O(\log n)$
- D. $O(n^2)$

Answer:

Question 7:

During add(), which operation restores the heap invariant?

- A. percolateDown()
- B. percolateUp()
- C. Rotating the tree
- D. Sorting the array

Answer:

Question 8:

In a heap's array representation, where is the root node stored?

- A. Index 0
- B. Index 1
- C. The middle of the array
- D. The last index

Answer:

Question 10:

Which traversal of a BST returns elements in sorted order?

- A. Pre-order
- B. In-order
- C. Post-order
- D. Level-order

Answer:

Question 11:

What is the primary advantage of using a heap over a BST for priority queues?

- A. Faster search for arbitrary elements

- B. Simpler implementation for maintaining min/max
- C. Guaranteed $O(1)$ time for all operations
- D. In-order traversal returns sorted data

Answer:

Question 12:

Which property distinguishes a heap from a BST?

- A. Heaps are always complete trees
- B. BSTs enforce a strict ordering invariant
- C. Heaps allow duplicates
- D. BSTs use arrays for storage

Answer:

Question 13:

When using Floyd's buildHeap, why is the time complexity $O(n)$ instead of $O(n\log n)$?

- A. Most nodes are near the bottom and require fewer swaps
- B. The algorithm uses a sorted array
- C. It ignores the heap invariants
- D. It skips the percolateDown step

Answer: