True, False, Explain

Decide whether each of the following statements is true or false, and give a reason.

Problem 1. Given an array of n distinct comparable integers, we can identify and sort the $\frac{n}{\log n}$ smallest of them in $O(n)$ time using a heap.

[True / False]

Solution. True. We can build a heap on array in $O(n)$ and pop the k largest in order in $O(klogn)$ time. For $k = n/\log n$, this running time is $O(n)$.

Problem 2. Given k distinct integer keys, there exists a binary search tree containing all k of them that satisfies the max heap property.

[True / False]

Solution. True. A chain to the left where each subtree is rooted at its max element.

Problem 3. Depth-first search solves single-source shortest paths in an unweighted, directed graph $G = (V,E)$ in $O(|V|+|E|)$-time.

[True / False]

Solution. False. A complete graph on three vertices is a counter example

Problem 4. Given a connected weighted directed graph having positive integer edge weights, where each edge weight is at most k, we can compute single source shortest paths in $O(k|E|)$ time.

[True / False]

Solution. True. Replace each edge (a,b) with weight w with a directed unweighted path from a to b, and run BFS from the source.

Problem 5. Given a Set AVL tree storing n keyed items ordered by key, one can construct a key-ordered max-heap on the same n items in worst-case $O(n)$ time.

[True / False]

Solution. True. Perform an in-order traversal of the AVL tree in $O(n)$ time to output an array of the items in the tree, and then transform the array into a max-heap (build the heap) in $O(n)$ time.

Problem 6. Given a weighted connected undirected graph $G = (V,E)$ containing exactly $|V|-1$ edges, one can solve weighted Single-Source Shortest Paths from any $s \in V$ in $O(|V|)$ time.

[True / False]

Solution. True. A connected graph with exactly $|V|-1$ edges is a tree and contains no undirected cycles. Run BFS or DFS from s in $O(|V|)$ time.

Problem 7. A max heap can be converted into a min heap in linear time.

[True / False]

Solution. True. Building a min heap on any array takes linear time.

Problem 8. Performing a single rotation on a binary search tree always results in binary tree that also satisfies the BST Property.

[True / False]

Solution. True. Rotations change the position and connections of nodes, but key order is maintained.

Problem 9. If a node a in an AVL tree is not a leaf, then a's successor is a leaf.

[True / False]

Solution. False. For example, consider the tree containing vertices a, b, c, d where:

c: left child a, right child d

a: left child b

Problem 10. Given an array of n integers representing a binary min-heap, one can find and extract the maximum integer in the array in $O(\log n)$ time.

[True / False]

Solution. False. The maximum element could be in any leaf of the heap, and a binary heap on n nodes contains at least $n/2$ leaves.

Problem 11. Any binary search tree on n nodes can be transformed into an AVL tree using $O(\log n)$ rotations.

[True / False]

Solution. False. Since any rotation changes height of any node by at most a constant, a chain of n nodes would require at least $n - \log n$ rotations.

Problem 12. Given a graph where all edge weights are strictly greater than -3, a shortest path between vertices s and t can be found by adding 3 to the weight of each edge and running Dijkstra's algorithm from s.

[True / False]

Solution. False. Counter example: a graph on vertices s, t, v, with undirected weighted edges $w(s, v) = 0$, $w(s, t) = -1$, and $w(v, t) = -2$.

Credits: Problems taken from MIT 6.006 Introduction to Algorithms