CSC263H Tutorial 2

Sample Solutions

Winter 2024

Work on these exercises *before* the tutorial. You don't have to come up with a complete solution, but you should be prepared to discuss them with your TA. We encourage you to work on these problems in groups of 2.

1. Alice claims that the minimum element of a binary max-heap must be one of its leaf nodes. Do you agree? Prove or disprove it.

   Answer: Yes. Let's prove the claim by contradiction. Suppose that there is no minimum element on a leaf node. Then, there must exist some node whose value is the minimum, and this node must have at least one child node whose value is no greater than the minimum element. Therefore, the child node also must have the minimum value, and hence cannot be a leaf node. Consequently, there must be an infinite chain of nodes that have the minimum value. However, this contradicts the fact that a binary max-heap is a finite graph.

2. Bob claims that the median element of a binary max-heap must be one of its leaf nodes. Do you agree? Prove or disprove it.

   Answer: No. Consider the binary max-heap (represented an array): $[100, 20, 25, 10, 5]$. The median value is 20, and it is not a leaf node in the heap since it has two child nodes, 10 and 5.

3. A ternary max-heap is like a binary max-heap except that non-leaf nodes have three children instead of two children. (As with binary heaps, there can of course be one non-leaf node that has fewer than three children.) We refer to the children as the left child, middle child, and right child.

   The values stored in the nodes are ordered according to the same principle as for binary max-heaps: the value at each node is greater than or equal to the values in the node's children. Answer the following questions regarding ternary max-heaps.

   (a) Similar to binary heaps, a ternary heap is represented by an array. Given the index i of an element in the array, what are the indices of the left child, the middle child, the right child, and the parent of the element? Assume that the array index starts at 0. Be thorough and discuss all possible corner cases.
   Answer:The left child is at $3(i+1)-2$, mid child is at $3(i+1)-1$, the right child is at $3(i+1)$. The parent location for left child node is $(i+2)/3-1$, mid child is $(i+1)/3-1$, and right child is $i/3-1$.

   (b) How do EXTRACT-MAX and INSERT work for a ternary max-heap? Explain their differences from the corresponding operations for binary max-heaps. No pseudo-code required.
   Answer: EXTRACT-MAX and INSERT operations on a ternary max-heap work similarly to those on a binary max-heap. EXTRACT-MAX first swaps the root with the right-most leaf node and then deletes the leaf node. The new root is propagated downwards to maintain heap consistency. INSERT adds a new node as the right-most leaf and then propagates the new node upwards.

(c) Consider a function IS-TERNARY-MAX-HEAP(A) that takes an array A as the input and returns TRUE if and only if A represents a valid ternary max-heap. Write the pseudo-code of a recursive implementation (i.e., no loops) of IS-TERNARY-MAX-HEAP. Briefly explain (not a proof) the correctness of your pseudo-code and give an asymptotic upper-bound (big-Oh) on the worst-case runtime of your pseudo-code.

Answer: assume helper functions $\mathrm{C}_1(i)$, $\mathrm{C}_m(i)$and $\mathrm{C}_r(i)$ exist that returns the left-child, mid-child and right-child of a node at index i.

IS-TERNARY-MAX-HEAP(A) = REC(A, 0) where

$$\text{REC(A, i)} = \begin{cases} \top & \text{if } A[i] \text{ is a leaf} \\ A[i] >= max(A[\mathrm{C}_1(i)], A[\mathrm{C}_m(i)], A[\mathrm{C}_r(i)]) \wedge & \\ \text{REC(A, } \mathrm{C}_1(i)) \wedge \text{REC(A, } \mathrm{C}_m(i)) \wedge \text{REC(A, } \mathrm{C}_r(i)) & \text{otherwise} \end{cases}$$

REC(A, i) checks if i is no less than all of its child nodes, and then recursively calls itself on all of its child nodes to ensure the max-heap invariant. The runtime is O(N), where N is the size of the heap.

(d) Write the pseudo-code of a iterative implementation (i.e., no recursion) of IS-TERNARY-MAX-HEAP. Briefly explain (not a proof) the correctness of your pseudo-code and give an asymptotic upper-bound (big-Oh) on the worst-case runtime of your pseudo-code.

Answer: assume helper functions $\mathrm{C}_1(i)$, $\mathrm{C}_m(i)$and $\mathrm{C}_r(i)$ exist that returns the left-child, mid-child and right-child of a node at index i.

IS-TERNARY-MAX-HEAP(A) = For $i \in [0, |A|]$ ASSERT $(A[i] >= max(A[\mathrm{C}_1(i)], A[\mathrm{C}_m(i)], A[\mathrm{C}_r(i)]))$

The iterative implementation loops over every element of the heap to check the max-heap invariant (i.e., the parent node is greater than its child nodes). The runtime complexity is O(N).