ShanghaiTech University

CS101 Algorithms and Data Structures

Fall 2021

Homework 5

Due date: 23:59, October 31, 2021

1. Please write your solutions in English.
2. Submit your solutions to gradescope.com.
3. Set your FULL Name to your Chinese name and your STUDENT ID correctly in Account Settings.
4. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
5. When submitting, match your solutions to the according problem numbers correctly.
6. No late submission will be accepted.
7. Violations to any of the above may result in zero grade.

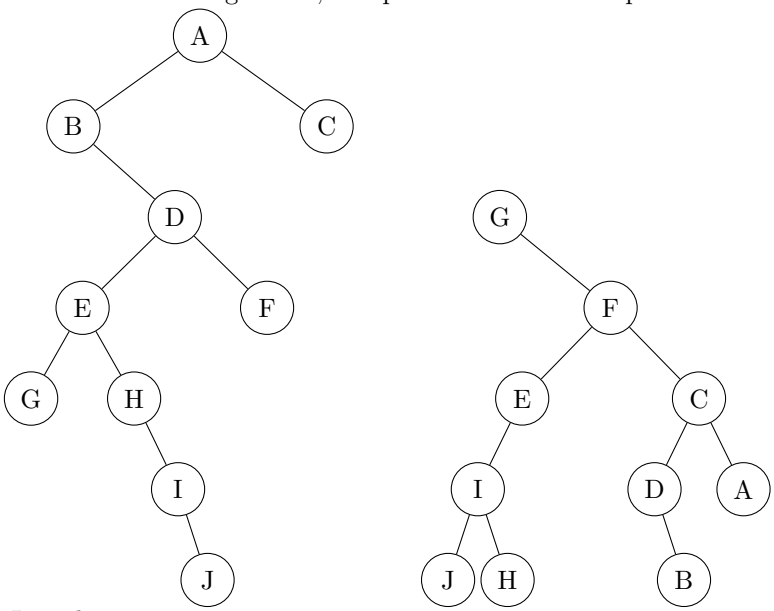
Problem 1: Multiple Choices

Multiple Choices: Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get half points if you select a non-empty subset of the correct answers.
Note that you should write you answers of Problem 1 in the table below.

Q(1)	Q(2)	Q(3)

(1) Which of the following are true about Binary Trees? (2pts)
A. Every full binary tree is also a complete binary tree.
B. A full binary tree with n nodes has height of $O(\ln n)$.
C. A complete binary tree with n leaves has exactly $2n - 1$ nodes.
D. A full binary tree with n leaves has exactly $2n - 1$ nodes.
E. None of the above.

(2) Which traversals of the left tree and right tree, will produce the same sequence node name? (2pts)



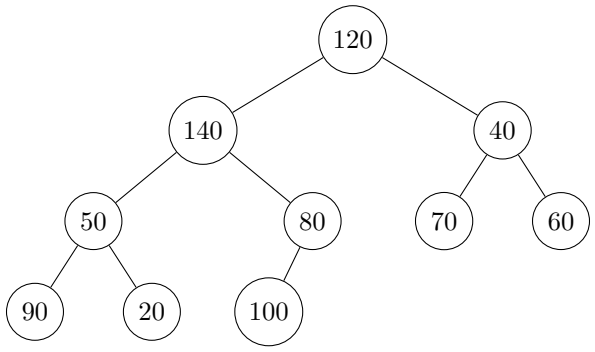
A. left: In-order, right: In-order
B. left: Pre-order, right: Pre-order
C. left: Post-order, right: Post-order
D. left: Post-order, right: In-order

(3) Which of the following statements about the binary heap is **not** true? Note that binary heaps mentioned in this problem are implemented by complete binary trees. (2pts)
A. There exists a heap with seven distinct elements so that the in-order traversal gives the element in sorted order.
B. If item A is an ancestor of item B in a heap (used as a Priority Queue) then it must be the case that the Insert operation for item A occurred before the **Insert** operation for item B.
C. If array A is sorted from smallest to largest then A (excluding A[0]) corresponds to a min-heap.

D. None of the above.

Problem 2: Construct a Heap

(1) Suppose we construct a min-heap from the following initial heap using Floyd's method. After the construction is completed, what should this heap look like? Please draw it as a binary tree. (5pts)

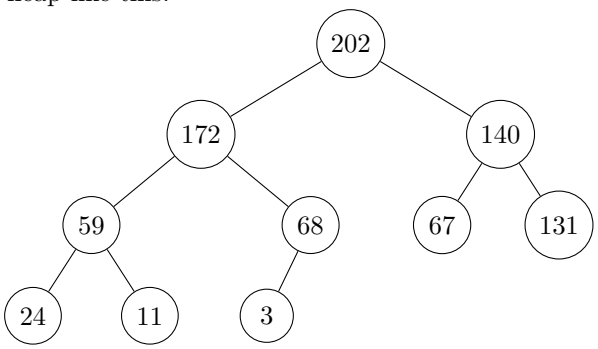


(2) Suppose we delete the root from the heap constructed in (1). What will be the post-order traversal of the new heap? (2pts)
Your answer: _____.

(3) Suppose there is a binary tree with the same post-order traversal of the heap in (2), and the binary tree has an **in-order** traversal of $120, 140, 80, 90, 40, 60, 50, 100, 70$. Please draw this binary tree below. (5pts)

Problem 3: Heap Sort

You are given such a max heap like this:



Then you need to use array method to show each step of heap sort in increasing order. Fill in the value in the table below. Notice that the value we have put is the step of each value sorted successfully. For each step, you should always make you heap satisfies the requirement of max heap property. (10pts)

index	0	1	2	3	4	5	6	7	8	9	10
value											

Table 1: The original array to represent max heap.

index	0	1	2	3	4	5	6	7	8	9	10
value											202

Table 2: First value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value										172	202

Table 3: Second value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value									140	172	202

Table 4: Third value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value								131	140	172	202

Table 5: Fourth value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value							68	131	140	172	202

Table 6: Fifth value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value						67	68	131	140	172	202

Table 7: Sixth value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value					59	67	68	131	140	172	202

Table 8: Seventh value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value				24	59	67	68	131	140	172	202

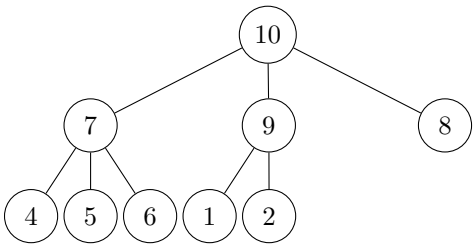
Table 9: Eighth value is successfully sorted.

index	0	1	2	3	4	5	6	7	8	9	10
value			11	24	59	67	68	131	140	172	202

Table 10: Last 2 values are successfully sorted.

Problem 4: i-Heap

In class, Prof. Zhang has mentioned the method of the array storage of a binary heap. In order to have a better view of heap, we decide to extend the idea of a binary heap to i-heap. In other words, each node in the heap now has at most i children instead of just two, which is stored in a complete binary tree.
For example, the following heap is a 3-heap.



(1) (4pts) If you are given the node with index x, what is the index of its parent and its yth child ($1 \leq y \leq i$)?
Notice: We assume the root is kept in $A[1]$. For your final answer, please represent it in terms of i, x and y. Please use flooring or ceiling to ensure that your final answer is as tight as possible if you need, i.e. $\lfloor m \rfloor$, $\lceil m \rceil$.

(2) (4pts) What is the height of a i-heap of n elements? Please show your steps.
Notice: For your final answer, please represent it in terms of i and n. Please use flooring or ceiling to ensure that your final answer is as tight as possible if you need, i.e. $\lfloor m \rfloor$, $\lceil m \rceil$.

(3) (4pts) Now we want to study which value of i can minimize the comparison complexity of heap-sort. For heap-sort, given a built heap, the worst-case number of comparisons is $\Theta(nhi)$, where $h = \Theta(\log_i n)$ is the height of the heap. Suppose the worst-case number of comparisons is $T(n, i)$. You need to do:

- Explain why $T(n, i) = \Theta(nhi)$.
- Suppose n is fixed, solve for i so that $T(n, i)$ is minimized.

Notice: i is an integer actually. In this problem, we only consider the complexity of comparison, not the accurate number of comparison.

(4) (5pts) TA Xiaozhang Master has a new idea that if $i = 1$, we only need to do the `BUILD-HEAP` operation to get a sorted array. Since we know from the lecture that `BUILD-HEAP` takes $O(n)$ time, he thinks it can actually sort in $O(n)$ time!

Now you are the student of TA Xiaozhang Master, and you need to judge his statement is true or false.

- If his statement is true, please explain the reason and prove its correctness.
- If his statement is false, please help him find the fallacy in his argument with your own reason. In the heap-sort he proposes, what sorting algorithm is actually performed? What is the worst running time of such a sorting algorithm?