

Problem 1(10pts): True or False: For each statement, choose T if the statement is correct, otherwise, choose F.

Fill all the answers in the table below.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
F	T	F	T	T	T	F	T	F	T

- (1) Every full binary tree is also a complete binary tree.
- (2) There are n elements inserted one by one into an initially empty binary heap. The total time complexity is $O(n \log n)$.
- (3) A full binary tree with n non-leaf nodes contains $2n - 1$ total nodes.
- (4) A binary tree of height $h = 0$ is perfect.
- (5) We have a binary heap of n elements and wish to add n more elements into it while maintaining the heap property. It can be done in $O(n)$.
- (6) In a binary max-heap containing n numbers, the smallest element can be found in time $O(n)$.
- (7) If a binary tree is a min-heap, then the post-order traversal of this tree is descending.
- (8) If the post-order traversal of a complete binary tree is descending, then the binary tree is a min-heap.
- (9) If the pre-order traversal and post-order traversal of two binary trees are equal respectively, then the two binary trees are exactly the same.
- (10) In a binary tree, every non-root node has exactly one parent.

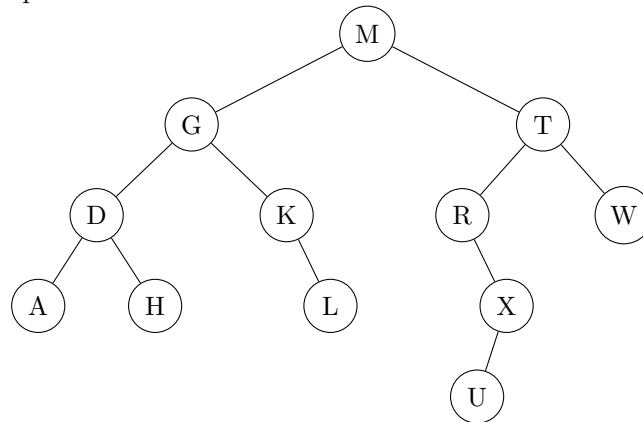
Problem 2(9pts) Fill in the blanks.

- (1) The maximum number of nodes in a binary tree of height h is _____ $2^{h+1} - 1$
- (2) In a binary min-heap with n elements, we can only access the top of the heap. The 7^{th} smallest element can be found in time $O(\text{_____})$ if duplicated elements are not allowed.
 $\log(n)$
- (3) In a binary min-heap with n elements, we can access the array storing the heap. The 7^{th} smallest element can be found in time $O(\text{_____})$ if duplicated elements are not allowed.
 1
- (4) (3pts) The elements 32, 15, 20, 30, 12, 25, 33 are inserted **one by one** in the given order into a max-heap. Please represent the final heap as an array (in-place).

0	1	2	3	4	5	6
33	30	32	15	12	20	25

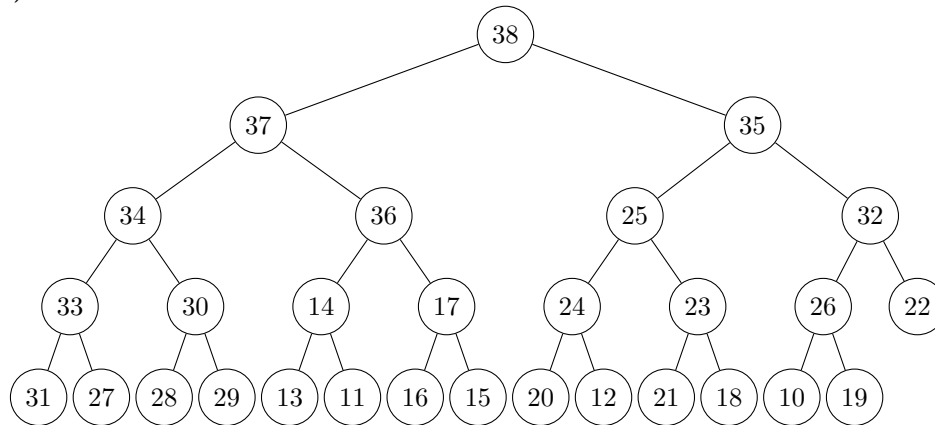
Problem 3(6pts) Given the in-order and post-order traversal of a binary tree T are $ADHGKLMRUXTW$ and $AHDLKGUXRWTM$ respectively.

Draw the tree T and write the pre-order traversal of T .



The pre-order of the tree is *MGDAHKLTRXUW*.

Problem 4(6pts)



(1) (1pt) Is this heap a max-heap or a min-heap?

max-heap

(2) (3pts) Suppose that you pop the key from the heap above. Write down all the elements that are involved in one (or more) compares.

14,17,19,34,35,36,37. The compares are 35-37, 19-37, 34-36, 19-36, 14-17, 19-17

(3) (3pts) Suppose that inserting the key x was the last operation performed in the binary heap in the figure. That is, after inserting x , the heap is shown as the figure above. Write down all possible value of x .

19,26,32,35. To insert a node in a binary heap, we place it in the next available leaf node and swim it up. Thus, 19,26,32,35, and 38 are the only keys that we might move. But, the last inserted key could not have been 38, because, then, 35 would have been the old root (which would violate heap order because the left child of the root is 37).