

**1. (8 points) True or False**

For each statement, choose T if the statement is correct, otherwise, choose F.

*Note: You should write down your answers in the box below.*

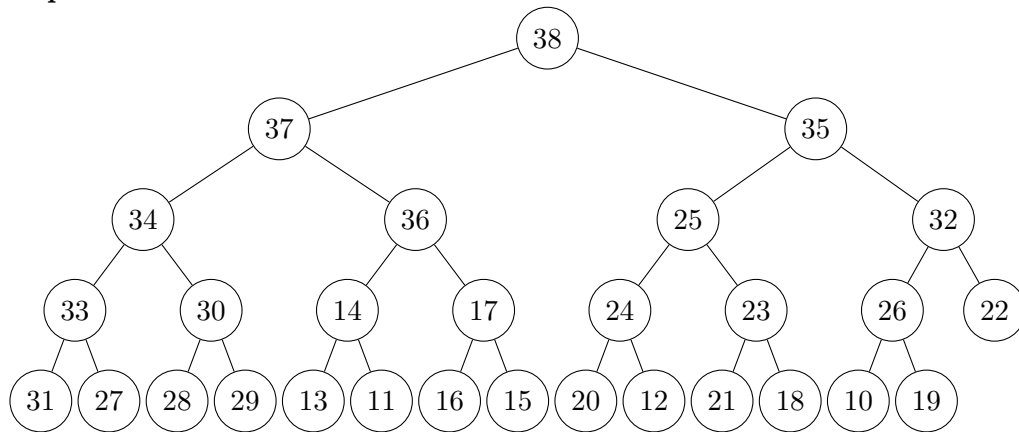
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)

- (a) (1') A binary tree of height  $h = 0$  is perfect.
- (b) (1') Every binary tree has at least one node.
- (c) (1') If the pre-order traversal and in-order traversal of two binary trees are equal respectively, then the two binary trees are exactly the same.
- (d) (1') BFS and DFS on a binary tree always give different traversal sequences.
- (e) (1') In a binary max-heap containing  $n$  numbers, the smallest element can be found in time  $O(n)$ .
- (f) (1') If the post-order traversal of a complete binary tree is descending, then the binary tree is a min-heap.
- (g) (1') Every complete binary tree has a perfect binary sub-tree.
- (h) (1') Every full binary tree is also a complete binary tree.

**2. (11 points) Fill-in-the-blanks**

- (a) (2') A full binary tree with  $n$  non-leaf nodes contains \_\_\_\_\_ total nodes.
- (b) (2') There are \_\_\_\_\_ distinct shapes of binary trees with 5 nodes.
- (c) (4') In a binary min-heap with  $n$  elements and duplicated elements are not allowed, the 7<sup>th</sup> smallest element can be found in time  $O(\text{_____})$  if we can only access the top of the heap ( $n \gg 7$ ). And the 7<sup>th</sup> smallest element can be found in time  $O(\text{_____})$  if we can access the array storing the heap.
- (d) (3') Suppose we have an initial heap stored in an array as (120, 140, 40, 50, 80, 70, 60, 90, 20, 100), we will construct a min-heap from the initial heap using Floyd's method. After the construction is completed, we delete the root from the heap. Then the post-order traversal of the heap will be \_\_\_\_\_.

3. (6 points) Heap



- (a) (1') Is this heap a max-heap or a min-heap?
- (b) (3') Suppose that you pop the key from the heap above. Write down all the elements that are involved in one (or more) compares.
- (c) (2') 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  $x$ .