---

1. Choose the appropriate running time from the list below.

The variable n represents the number of items (keys, data, or key/data pairs) in the structure. In answering this question you should assume the best possible implementation given the constraints, and also assume that every array is sufficiently large to handle all items (unless otherwise stated).

Perform an In-order traversal of a Binary Tree.

   A. $O(n \log n)$
   B. [Correct Answer] [Your Answer] $O(n)$
   C. $O(1)$
   D. $O(\log n)$
   E. $O(n^2)$

---

2. Consider the binary tree class described in lecture where we have 1) variable `root` that is the `treeNode` representing the root of the binary tree and 2) each `treeNode` consists of an integer `data` element, and two `treeNode` pointers called `left` and `right`.

What does fun(`root`) return?

```
int fun(treeNode * curr) {
    if (curr != null) {
        ret1 = fun(curr->left);
        ret2 = fun(curr->right);
        return 1 + ret1 + ret2;
    }
    else return 0;
}
```

   A. `fun` returns the shortest distance from root to leaf.
   B. `fun` returns the height of the tree.
   C. [Correct Answer] `fun` returns the number of elements in the tree.
   D. [Your Answer] None of the other options is correct.
   E. `fun` returns the sum of all elements in the tree.

---

3. Suppose a binary tree holds 357 keys. Then our node-based implementation of that tree has how many `NULL` pointers?

   A. The answer cannot be determined from the information given.
   B. 712
   C. [Correct Answer] [Your Answer] 358
   D. None of these is the correct response.
   E. 178

---

4. Among the following choices, which abstract data type should be used for a level order traversal of a binary tree?

   A. [Correct Answer] [Your Answer] queue
   B. array
   C. stack
   D. linked list
   E. hash table

---

5. Choose the appropriate running time from the list below.

The variable n represents the number of items (keys, data, or key/data pairs) in the tree and h represents the height of the tree. In answering this question you should assume the best possible implementation given the constraints, and also assume that every array is sufficiently large to handle all items (unless otherwise stated).

Given a *perfect* binary tree, compute the length of the longest path from v down to a descendant leaf.

   A. $O(1)$
   B. $O(n^2)$
   C. None of the options is correct
   D. [Correct Answer] [Your Answer] $O(h)$
   E. $O(n)$