Algorithms and Data Structures

Valdrin Smakaj

Answers Sheet-9

9.2)

a) Since we are still using a bit of memory for the pointer variables]------>
it is still considered as because it stays in the same constant value]------>
no matter how large the input can be, so no changes or larger uses of]------>
memory occur in this case, therefore it is an in situ algorithm.]------>

b) Best case: T(n) = 2T(n/2) + O(n) -> O(n log n).
Reason: Two recursive calls (T(n/2)) and the push_back function (O(n))

Worst case: T(n) = T(n-1) + O(n) -> O(n^2).
Reason: First recursive call (T(n-1)), and no other since the tree has only left or right side.

c) Searching for an element in the end of the tree for instance,
takes time complexity of the height of the tree(root->leaf)
Therefore -> O(h).

In the tree perspective, searching will be done in linear time O(n)

```
void reverse(){
    Node* current = head;
    Node *prev = NULL, *next = NULL;

    while (current != NULL) {
        // Store next
        next = current->next;
        // Reverse current node's pointer
        current->next = prev;
        // Move pointers one position ahead.
        prev = current;
        current = next;
    }
    head = prev;
}
```