TASK 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Unsorted,singly linked | Sorted, singly linked | Unsorted,doubly linked | Sorted, doubly linked |
| Search(L,k) | O(n) | O(n) | O(n) | O(n) |
| Insert(L,x) | O(1) | O(n) | O(1) | O(n) |
| Delete(L,x) | O(n) | O(n) | O(1) | O(1) |
| Successor(L,x) | O(n) | O(1) | O(n) | O(1) |
| Predecessor(L,x) | O(n) | O(n) | O(n) | O(1) |
| Minimum(L) | O(n) | O(1) | O(n) | O(1) |
| Maximum(L) | O(n) | O(n) | O(n) | O(1) |

TASK 2

In book’s given pseudocode, started from line 5, everywhere written *left* becomes *right,* and every *right* becomes *left.* For fair strategy we can just randomly generate successor or predecessor is going to be used.

TASK 3

The logic here is that when we are looking at any x element in the given array, and if x’s next element is greater(less) than x, all the upcoming ones must be greater(less) than x, because if it is greater(less) it means that now we are looking for the searched number in the right(left) subarray, where all the elements are we are greater(less) than the parent node.

c. 925, 202, 911, 240, 912, 245, 363

In the array after 911 is coming 240, which is less than 911, so now we are looking for 363 in the left subtree, which means that all the upcoming elements must be less than 911, but somehow we see 912 in array, which is greater than 911.So the logic in C is broken.

e. 935, 278, 347, 621, 299, 392, 358, 363

In the array after 347 is coming 621, which is greater than 347, so now we are looking for the searched node in the right subtree, which means that all the upcoming elements must be greater than 347, but somehow we see 299 in array, which is less than 347.So the logic in E is broken.

The answer is C and E.