

1. Consider a class `List` that is implemented using a doubly linked list with `head` and `tail` pointers (i.e. pointers to the first and last nodes in the list).

Given that implementation, which of the following operations could be implemented in $O(1)$ time?

- I. Insert item at the front of the list
 - II. Insert item at the rear of the list
 - III. Delete front item from list
 - IV. Delete rear item from list
- A. I, II and IV
 - B. I, II and III
 - C. I and II
 - D. **[Correct Answer]** **[Your Answer]** All of them
 - E. I and III

2. Which of the following List ADT implementations gives us an $O(1)$ time for `insertAtFront`, i.e inserting an element at the front of the list?

- I. A singly-linked list with only a `head` pointer.
 - II. A singly-linked list with `head` and `tail` pointers.
 - III. A doubly-linked list with only a `tail` pointer.
 - IV. A doubly-linked list with `head` and `tail` pointers.
- A. **[Correct Answer]** **[Your Answer]** I, II and IV
 - B. II and IV
 - C. I and II
 - D. None of the other options is correct
 - E. I, II, III and IV

3. In a doubly linked list, what will be the time required to insert at the middle position of the list?

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

4. Consider the following function definition and suppose that 1) the `node` class consists of an integer data element, and a node pointer called `next`, and 2) variable `head` is the address of a linked list of such nodes.

What does the function do?

```
void fun(node * curr) {
    if (curr != NULL)
        cout << curr->data;
        if (curr->next != NULL) {
            fun(curr->next->next);
        }
}

node * head = NULL;
// maybe insert data into the chain here
fun(head);
```

- A. `fun` prints the elements of the list from `head` to the end.
- B. **[Correct Answer]** **[Your Answer]** `fun` prints every other element of the list.
- C. None of the other options is correct.
- D. **[Your Answer]** `fun` prints the reverse of the list.
- E. `fun` segfaults on lists of odd length.

5. In a singly linked list containing n nodes, the time required to find the maximum element is:

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