Jabir Chowdhury

## 1.1

```
int length (Node *l) {
    Node * cur = l;
    int i = 0;
    While ( cur != nullptr){
        i++;
        cur = cur -> getNext();
    }
    return i;
}
```

## 1.2

```
Node *middle (Node * head) {
    int mid_ind = length(head)/2 ;
    Node *cur = head;
    While (mid_ind != 0) {
        Cur = cur -> getNext();
        mid_ind--;
    }
    return cur;
}
```

Scanned with CamScanner

1.3

```
Node * msort (Node *head) {
    int len = length(head);
    if ( len <= 1) return head;

    Node * a;
    Node * b;

    b = middle(head);
    a = head;
    int mid_ind = length(head) /2;
    Node * cur = head;
    While (mid_ind != 1) {
        cur = cur->getNext();
        mid_ind --;
    }
    cur->setNext (nullptr);

    Node * left= msort (a);
    Node * right = msort (b);

    Node * ret = merge (left, right);
    return ret;
}
```

1.4

```
Node * rotate (Node *head, int position) {
        Node * cur = head;
        int pos = position;
        if (pos == 1) return head;
        while (pos != 2) {
             cur = cur -> getNext();
             pos--;
        }
        Node * next = cur -> getNext();
        cur -> setNext (nullptr);

        Node * temp = next;
        while (temp -> getNext() != nullptr) {
               temp = temp -> getNext ();
        }
        temp -> setNext (head);
        head = next;
        return next;
}
```

## 2

2.1) $O(n)$

2.2) $O(c)$

2.3) $O(1)$

2.4) $O(\log n)$

2.5) $O(1)$

2.6) $O(n)$