CSC 212 Midterm 1 Solution - Spring 2014

College of Computer and Information Sciences, King Saud University
Exam Duration: 2 Hours

12/03/2014

Question 1 [25 points]

1.

	Statement	S/E	Frequency	Total
1	void func(int n) {	0	-	0
2	for(int i=n-1; i >= 0; i=i-1) {	1	$n+1$	$n+1$
3	int sum= 0;	1	n	n
4	for(int j=0; j <i; j++) {	1	$n(n+1)/2$	$n(n+1)/2$
5	sum+= j;	1	$n(n-1)/2$	$n(n-1)/2$
6	System.out.println(sum);	1	$n(n-1)/2$	$n(n-1)/2$
7	}	0	-	0
8	}	0	-	0
10	}	0	-	0
	Total operations	$3/2n^2+3/2n+1$		
	Big-oh	$O(n^2)$		

2.

	Statement	S/E	Frequency	Total
1	void func(int n) {	0	-	0
2	for(int i=0; i<n*n; i++) {	1	n^2+1	n^2+1
3	int j=1;	1	n^2	n^2
4	while(j <= n) {	1	n^3+n^2	n^3+n^2
5	j++ ;	1	n^3	n^3
6	System.out.println(i+j);	1	n^3	n^3
7	}	0	-	0
8	}	0	-	0
10	}	0	-	0
	Total operations	$3n^3+3n^2+1$		
	Big-oh	$O(n^3)$		

Question 2 [25 points]

1.

2.

```
public void removeLast() {
        if (head.next == null) {
                current = head = null;
                return;
        }
        Node<T> p = head;
        while (p.next != null)
                p= p.next;
        p.previous.next = null;
        if (current == p)
                current = head;
}
```

Question 3 [25 points]

1.

```
public <T> void replace(Queue<T> q, int i, int j) {
        T val;
        for(int k = 0; k < q.length(); k++) {
                if (k == j) {
                        val = q.serve();
                        q.enqueue(val);
                }
                else {
                        q.enqueue(q.serve());
                }
        }
        for(int k = 0; k < q.length(); k++) {
                if (k == i) {
                        q.serve();
                        q.enqueue(val);
                }
                else {
                        q.enqueue(q.serve());
                }
        }
}
```

2.

```
public void replace(int i, int j) {
        Node<T> pi = head;
        for(int k = 0; k < i; k++)
                pi = pi.next;
        Node<T> pj = head;
        for(int k = 0; k < j; k++)
                pj = pj.next;
        pi.data = pj.data;
}
```

3. Both methods are $O(n)$, hence they have the same performance.

Question 4 [25 points]

1.

```
public static <T> void insertIth(List<T> l, int i, T e) {
        l.findFirst();
        for(int k = 0; k < i; k++)
                l.findNext();
        l.insert(e);
}
```

2.

```
public void reverse() {
        Node<T> q = null;
        Node<T> p = head;
        while (p != null) {
                Node<T> tmp = p.next;
                p.next = q;
                q = p;
                p = tmp;
        }
        head = q;
}
```