CSC 212 Midterm 1 solution - Fall 2017

College of Computer and Information Sciences, King Saud University
Exam Duration: 90 Minutes

9/11/2017

Question 1 [30 points]

(1) Code analysis:

Line	Frequency
1	d []
2	a []
3	e []
4	b []
Big oh	c []

(2) Code analysis:

Line	Frequency
2	d []
3	b []
4	d []
5	a []
6	d []
7	c []
Big oh	c []

Question 2 [35 points]

(1)

```
public static <T> void removeAllX(List<T> l, T[] X, int n)
{
if(l.empty() || n==0) return;

for (int i=0;i<n;i++)
{
l.findFirst();

while(!l.last())
{
if (X[i].equals(l.retrive()))
l.remove();

else
l.findNext();
}
//last element
if (X[i].equals(l.retrive()))
l.remove();
}


}
```

(2)

```
public static <T> void reverse(Queue<T> q){

DoubleLinkedList<T> tmp = new DoubleLinkedList<T>();
int n=q.length();
for (int i=0; i<n; i++)
tmp.insert(q.serve());

while (!tmp.first()) {
q.enqueue(tmp.retrive());
tmp.findPrevious();
}
q.enqueue(tmp.retrive());

}
```

Question 3 [35 points]

(1)

```
public void moveToEnd(int p)
{
T e=null;
//move current to last node
while (current.next!=null)
current=current.next;
for (i=0;i<p;i++)
{
e=head.data;
head=head.next;
current.next=new Node<T>(e);
current=current.next;
}
current=head;
}
public void moveToEnd(int p)
{
Node<T> target= head;
Node <T> prev=null;
Node <T> start=head;

for (i=0;i<p;i++)
{
prev=target
target=target.next;
}
head=target;
prev.next=null;

while (target.next!=null)
{
target=target.next;
}

target.next=start;
current=head;
}

public void moveToEnd(int p)
{
Node<T> temp= head;
Node <T> prev=null;
for (i=0;i<p;i++)
{
prev=head
head=head.next;
}
prev.next=null;
while (current.next!=null)
current=current.next;
current.next=temp;
current=head;
}
```

(2)

```
private void shiftSeg(Node<T> startN, Node<T> endN, int n)
{

if(endN.next==null && startN.previous==null)
return;
if(endN.next==null)
return;
if (n==0)
return;
/* All above cases 1pt*/

/*Cutting the segment 6pts*/
Node<T> temp;
if(startN.previous!=null)
startN.previous.next=endN.next;
else  //startN point to head
head=endN.next;

if(endN.next!=null)
endN.next.previous=startN.previous;

/* finding the correct position 6pts */
Node<T> temp=endN;
for(int i=0;i<n;i++)
{if(temp.next==null)
break;
temp=temp.next;}

//temp point to target node or last node
/*put the segment in the correct place 5pts */
startN.previous=temp;
endN.next=temp.next

if(temp.next!=null)
temp.next.previous=endN;
temp.next=startN;


}
```