

Boris Aygildiz 1901042252 CSE 222 Final Examination B. Aygildiz  
I hereby pledge on my honor that I will strictly adhere to academic integrity codes and the work done on this examination is solely my own and I will not receive/give any help from/to anybody or source during this examination.

1) Worst case occurs when the linked-list is sorted in the reverse order. In that case we compare current element with all the elements in the right side of the current.

In the worst case:

For loop takes,  $\Theta(n)$  time  
inner while loop takes,  $\Theta(n)$  time  
inside while loop; setting and getting takes,  $\Theta(n)$  time  
 $\Rightarrow O(n^3)$

part-2: I'd put all the elements inside an arraylist

```
ArrayList<E> arr = new ArrayList<E>();  
Iterator<E> iter = list.iterator();  
while (iter.hasNext()) {  
    arr.add(iter.next());  
}
```

$\Rightarrow O(n^2)$

```
for (int f = 1; f < arr.size(); f++) {
```

```
    int i = f - 1;
```

```
    E current = arr.get(f);
```

```
    while ((i > -1) && (arr.get(i).compareTo(current) == 1)) {
```

```
        arr.set(i + 1, arr.get(i));
```

```
        i--;
```

```
    }
```

```
    arr.set(i + 1, current);
```

```
    }
```

```
LinkedList<E> temp = new LinkedList<E>();
```

```
for (int i = 0; i < arr.size(); i++) temp.add(arr.get(i));
```

putting  
all the  
element  
in a linked list

2) First I'd put starting node inside a hashset and a queue. Then I'd run a for loop  $(n-1)$  times and put the neighbors of all the elements inside the queue. When I found all the elements of a single vertex I'd poll the element in the queue. For the neighbours I found in the last iteration, I'd check if they are inside the hashset (to check if they're visited before). If they are not I would save those vertices inside an array and return that array.

Adding an element to hashset takes  $\Theta(1)$  time

Getting the iterator for A-L takes  $\Theta(1)$  time

" " " " A-M takes  $\Theta(n)$  time

Since we have two nested functions and calling adj. iterator

For A-L it would take  $O(n^2)$  time

" A-M " " "  $O(n^3)$  time