C S 272/463 Introduction to Data Structures

Lab 6: Running time Analysis

I. Requirements

Please ANALYZE the worst-case running time of the following methods, WRITE down your analysis in DETAIL, and denote their time complexity in Big-O.

Hint: You need to define n first, before showing whether the method is O(n), O(logn), O(n2), etc. Please put your analysis to a word file.

The IntArrayBag has two instance variables.

```
public class IntArrayBag
{
        // Invariants of the IntArrayBag class:
        // 1. The actual number of elements in the bag is in the instance variable
        // manyItems, which is no more than data.length.
        // 2. For an empty bag, we do not care what is stored in data array;
        // for a non-empty bag, the elements in the bag are stored in data[0]
        // through data[manyItems-1], and we don not care what is in the
        // rest of the data array.
        private int[ ] data;
        private int manyItems;
        //methods
}
```

n = how many items are in the bag currently.

(1) (25 pts) The add method in IntArrayBag that we discussed in our class.

```
public void add(int element)
{
        if (manyItems == data.length)                          O(1)
        {
                int biggerArray[ ];                            O(1)
                biggerArray = new int[manyItems*2 + 1];        O(4) → O(1)
                for(int i=0;i < manyItems;i++){                n · 2k+1 → O(n)
                        biggerArray [i] = data[i];             O(1)
                }
                data = biggerArray;                            O(1)
        }
        data[manyItems] = element;                             O(1)
        manyItems++;                                           O(1)
}
```

The method is O(n)

(2) (25 pts) A method to count the number of occurrences of a particular element target. This method is implemented in the IntArrayBag class that we discussed in class.

```
public int countOccurrences(int target)
{
      int answer = 0 ;
      int index;
      answer = 0;
      for (index = 0; index < manyItems; index++)
            if (target == data[index])
            answer++;
      return answer;
}
```

n = number of elements in the bag, manyItems.

O(1)
O(1)
O(1)
n · 2k + 1 → O(n)
O(1)
O(1)
O(1)

The method complexity is O(n)

(3) (25 pts) A method to find a node at a specified position in a linked list starting from the given head. This method is implemented in the IntNode class that we discussed in class.

```
public static IntNode listPosition(IntNode head, int position)
{
      IntNode cursor;
      int i;
      if (position <= 0)
            throw new IllegalArgumentException("position is not
      positive");
      cursor = head;
      for (i = 1; (i < position) && (cursor != null); i++)
            cursor = cursor.link;
      return cursor;
}
```

n = number of nodes in the list.

O(1)
O(1)
O(1)
O(1)
n · 4k + 1 → O(n)
O(1)
O(1)

The method is O(n).

(4) (25 pts) A method to compute the number of nodes in a linked list starting from the given head. This method is implemented in the IntNode class that we discussed in class.

```
public static int listLength(IntNode head)
{
      IntNode cursor = null;
      int answer = 0;
      for (cursor = head; cursor != null; cursor = cursor.link)
            answer++;
      return answer;
}
```

O(1)
O(1)
n · 2k + 1 → O(n)
O(1)
O(1)

The method is O(n).

II. Submission

Submit through canvas a zipped file containing your word file.

III. Grading Criteria

The score allocation is beside the questions.