

**0** Overview**Instructions**

**Due Date: Sunday, Sept 29 @ 11:59PM**

**Total points: 50**

Your written solution may be either handwritten and scanned, or typeset. Either way, you must produce a PDF that is legible and displays reasonably on a typical PDF reader. This PDF should be submitted via autolab as WA2. You should view your submission after you upload it to make sure that it is not corrupted or malformed. Submissions that are rotated, upside down, or that do not load will not receive credit. Illegible submissions may also lose credit depending on what can be read. Ensure that your final submission contains all pages.

**You are responsible for making sure your submission went through successfully.**

**Written submissions may be turned in up to one day late for a 50% penalty.**

**No grace day usage is allowed.**

## 1 Part 1: Sorted Lists (and Hinted Access)

Each of the following questions ask about runtimes related to the `SortedList` class you defined in PA1. When answering, give the most specific runtime bound you can: If a Big Theta bound exists, give that. Otherwise, give the tightest upper bound (Big- $O$ ) you can. All bounds should be given in terms of  $n$  (The variable, as it appears in each question). In at most *two* sentences, explain the runtime you gave. **Answers without explanations will receive no credit.**

### Question 1 (4 pt)

What is the runtime of `SortedList.insert` on a list where `list.size() == n`.

### Question 2 (4 pt)

What is the runtime of the following snippet of code:

```
1 SortedList<Integer> list = new SortedList<>();
2 for (int i = 0; i < n; i++){
3     list.insert(i)
4 }
```

### Question 3 (4 pt)

What is the runtime of the following snippet of code:

```
1 SortedList<Integer> list = new SortedList<>();
2 LinkedListNode<Integer> hint = list.insert(0);
3 for (int i = 1; i < n; i++){
4     hint = list.insert(i, hint)
5 }
```

### Question 4 (4 pt)

What is the runtime of the following snippet of code:

```
1 SortedList<Integer> list = new SortedList<>();
2 for (int i = n; i >= 0; i--){
3     hint = list.insert(i)
4 }
```

### Question 5 (4 pt)

What is the runtime of the following snippet of code:

```

1 SortedList<Integer> list = new SortedList<>();
2 Random rand = new Random();
3 for (int i = 0; i < n; i++){
4     list.insert(rand.nextInt(n));
5 }

```

## 2 Part 2: Duplicate Values

The following questions have to do with the following function, which takes a `SortedList` of `Integers`, and a `LinkedListNode` and 'increments' the value of *one* element stored at the node to the next higher value. All questions assume that `list` is a `SortedList` where `list.size() == n`.

```

1 public void increment(SortedList<Integer> list,
2                       LinkedListNode<Integer> node)
3 {
4     LinkedListNode<Integer> ret =
5         this.insert(node.value+1, node);
6     this.remove(node);
7     return ret;
8 }

```

### Question 6 (2 pt)

Give the tight, unqualified upper bound on the runtime (Big- $O$ ) of `increment` on a list with `n` elements in it?

### Question 7 (2 pt)

Give the tight, unqualified upper bound on the runtime (Big- $O$ ) of the following code snippet. Assume that `node` is an arbitrary node of `list`.

```

1 for(int i = 0; i < n; i++){
2     node = list.increment(node)
3 }

```

### Question 8 (3 pt)

Does your answer to Q.6 change if the `SortedList` stores one `LinkedListNode` for each distinct element of the list, instead of grouping elements with the same value into a single node? In at most *two* sentences explain why or why not.

**Question 9** (3 pt)

Does your answer to Q.7 change if the `SortedList` stores one `LinkedListNode` for each distinct element of the list, instead of grouping elements with the same value into a single node? In at most *two* sentences explain why or why not.

**3** Part 3: Asymptotic Analysis

For each of the following growth functions, find specific values for constants  $c$  and  $N_0$  that will prove the requested bounds (or show that no such constants exist). Show all work for your proof. **Answers given without valid work will receive no credit.**

**Question 10** (5 pt)

$$\text{Let } f(n) = 6n + 5n \log(n^{10}) + 9$$

Prove that  $f(n) \in O(n \log(n))$

**Question 11** (5 pt)

$$\text{Let } g(n) = 2^{\log(n)} + 6n^4 + 3n^2$$

Prove that  $g(n) \in O(n^3)$

**Question 12** (10 pt)

$$\text{Let } h(n) = 6n^4 + 2n^2 + 10n$$

Prove that  $h(n) \in \Theta(n^4)$

## 4 Summations

1.  $\sum_{i=j}^k c = (k - j + 1)c$
2.  $\sum_{i=j}^k (cf(i)) = c \sum_{i=j}^k f(i)$
3.  $\sum_{i=j}^k (f(i) + g(i)) = \left(\sum_{i=j}^k f(i)\right) + \left(\sum_{i=j}^k g(i)\right)$
4.  $\sum_{i=j}^k (f(i)) = \left(\sum_{i=\ell}^k (f(i))\right) - \left(\sum_{i=\ell}^{j-1} (f(i))\right)$  (for any  $\ell < j$ )
5.  $\sum_{i=j}^k f(i) = f(j) + f(j+1) + \dots + f(k-1) + f(k)$
6.  $\sum_{i=j}^k f(i) = f(j) + \dots + f(\ell-1) + \left(\sum_{i=\ell}^k f(i)\right)$  (for any  $j < \ell \leq k$ )
7.  $\sum_{i=j}^k f(i) = \left(\sum_{i=j}^{\ell} f(i)\right) + f(\ell+1) + \dots + f(k)$  (for any  $j \leq \ell < k$ )
8.  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$
9.  $\sum_{i=0}^k 2^i = 2^{k+1} - 1$   
 $n! \leq c_s n^n$

## 5 Inequalities

1.  $f(n) \leq g(n)$  is true if you can find some  $h(n)$  where  $f(n) \leq h(n)$  and  $h(n) \leq g(n)$
2.  $f(n) \leq g(n)$  is true if you can find some  $h(n)$  where  $f(n) - h(n) \leq g(n) - h(n)$
3.  $f(n) \leq g(n)$  is true if you can find some  $h(n) \geq 0$  (for all  $n$ ) such that  $f(n) \cdot h(n) \leq g(n) \cdot h(n)$
4. Take it as a given that:  $\theta(1) \leq \theta(\log(n)) \leq \theta(n) \leq \theta(n^2) \leq \theta(n^k)$  (for  $k > 2$ )  $\leq \theta(2^n)$

## 6 Logarithms

1.  $\log(n^a) = a \log(n)$
2.  $\log(an) = \log(a) + \log(n)$
3.  $\log\left(\frac{n}{a}\right) = \log(n) - \log(a)$

4.  $\log_b(n) = \frac{\log_c(n)}{\log_c(b)}$

5.  $\log(2^n) = 2^{\log(n)} = n$