## 0 Overview

### Instructions

**Due Date: Sunday Feb 23 @ 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 may submit as many times as you like, but only your last submission will be graded and should include the entire submission. 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 or incomplete submissions will 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. You will only receive partial credit if your answer is not as specific as possible. 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. If the runtime depends on specific details of **your** implementation of PA1 make sure to include that in your explanation. **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:

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

### Question 3 (4 pt)

What is the runtime of the following snippet of code:

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

### Question 4 (4 pt)

What is the runtime of the following snippet of code:

```
SortedList<Integer> list = new SortedList<>();
list.insert(n);
list.insert(0);
for (int i = n - 1; i > 0; i--){
  list.insert(i, list.lastNode.get());
}
```

### Question 5 (4 pt)

What is the runtime of the following snippet of code (you may assume `new Random()` and `rand.nextInt(n)` run in $\Theta(1)$):

```
SortedList<Integer> list = new SortedList<>();
Random rand = new Random();
for (int i = 0; i < n; i++){
  list.insert(rand.nextInt(n));
}
```

## 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 'decrements' the value of *one* element stored at the node to the next lower value. All questions assume that `list` is a `SortedList` where `list.size() == n`.

```
public LinkedListNode<Integer> decrement(
    SortedList<Integer> list, LinkedListNode<Integer> node) {
  LinkedListNode<Integer> ret;
  ret = list.insert(node.value-1, node);
  list.remove(node);
  return ret;
}
```

### Question 6 (2 pt)

Give the tight, unqualified upper bound on the runtime (Big-$O$) of `deccrement` 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`.

```
for(int i = 0; i < n; i++) {
  node = list.decrement(node)
}
```

### 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. (If it would change your answer, include the new runtime in your explanation)

### 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)

Let $f(n) = 14 + 3n + 2^{\log(n^2)}$

Prove that $f(n) \in O(n^2)$

### Question 11 (5 pt)

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

Prove that $g(n) \in \Omega(n^2)$

### Question 12 (10 pt)

Let $h(n) = 5n^2 + 2^{n+4} + 1000$

Prove that $h(n) \in \Theta(2^n)$

## Summation Rules

S1. $\sum_{i=j}^{k} c = (k - j + 1)c$

S2. $\sum_{i=j}^{k}(cf(i)) = c\sum_{i=j}^{k} f(i)$

S3. $\sum_{i=j}^{k}(f(i) + g(i)) = \left(\sum_{i=j}^{k} f(i)\right) + \left(\sum_{i=j}^{k} g(i)\right)$

S4. $\sum_{i=j}^{k}(f(i)) = \left(\sum_{i=\ell}^{k}(f(i))\right) - \left(\sum_{i=\ell}^{j-1}(f(i))\right)$ **(for any $\ell < j$)**

S5. $\sum_{i=j}^{k} f(i) = f(j) + f(j+1) + \ldots + f(k-1) + f(k)$

S6. $\sum_{i=j}^{k} f(i) = f(j) + \ldots + f(\ell - 1) + \left(\sum_{i=\ell}^{k} f(i)\right)$ **(for any $j < \ell \leq k$)**

S7. $\sum_{i=j}^{k} f(i) = \left(\sum_{i=j}^{\ell} f(i)\right) + f(\ell + 1) + \ldots + f(k)$ **(for any $j \leq \ell < k$)**

S8. $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$

S9. $\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$
$n! \leq c_s n^n$

## Log Rules

L1. $\log(n^a) = a\log(n)$

L2. $\log(an) = \log(a) + \log(n)$

L3. $\log(\frac{n}{a}) = \log(n) - \log(a)$

L4. $\log_b(n) = \frac{\log_c(n)}{\log_c(b)}$

L5. $\log(2^n) = 2^{\log(n)} = n$

## Inequality Rules

I1. $f(n) \leq g(n)$ is true if you can find some $h(n)$ where $f(n) \leq h(n)$ and $h(n) \leq g(n)$

I2. $f(n) \leq g(n)$ is true if you can find some $h(n)$ where $f(n) - h(n) \leq g(n) - h(n)$

I3. $f(n) \leq g(n)$ is true if you can find some $h(n) \geq 0$ (for all $n$) such that $f(n) \cdot h(n) \leq g(n) \cdot h(n)$

I4. Take it as a given that: $\theta(1) \leq \theta(\log(n)) \leq \theta(n) \leq \theta(n^2) \leq \theta(n^k)$ **(for $k > 2$)** $\leq \theta(2^n)$