Written Assignment #2

Due Date: Sunday, Feb 22 @ 11:59PM
Total Points: 50

Instructions

Answer all questions in this written assignment, showing work when required. All work must be your own, created solely by you and using only the allowed resources for the course (as stated in the syllabus). Your solutions may either be handwritten and scanned, or typeset. Submit your work as a PDF via AutoLab to the WA2 submission target.

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.

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-Θ bound exists, give the Big-Θ bound.
- Otherwise, give **both** the tight upper bound (Big-O) and tight lower bound (Big-Ω).

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 [*3 points*]: What are the runtime bounds of `SortedList.findRef` when `list.length == n`, if no hint is given?

Question 2 [*3 points*]: What are the runtime bounds of `SortedList.findRef` when `list.length == n`, if a hint is given?

Question 3 [*4 points*]: What is the runtime of the following snippet of code? You should assume that `list` is an instance of `SortedList` with a length of `n`, containing values in the range `[1,n]`. The values are not guaranteed to be distinct, and therefore not every number in `[1,n]` is guaranteed to appear in the list.

```java
for (int i = 1; i <= n; i++) {
    list.findRef(i);
}
```

Question 4 [*5 points*]: What is the runtime of the following snippet of code? You should assume that `list` is an instance of `SortedList` with a length of `n`, containing values in the range `[1,n]`. The values are not guaranteed to be distinct, and therefore not every number in `[1,n]` is guaranteed to appear in the list.

```java
LinkedListNode<Integer> hint = list.headNode.get();
for (int i = 1; i <= n; i++) {
    Optional<LinkedListNode<Integer>> tmp = list.findRef(i, hint);
    if (tmp.isPresent()) hint = tmp.get();
}
```

Part 2: Duplicate Values

The following questions have to do with the following function, which takes a `SortedList<Integer>` 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`.

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

Question 5 [*3 points*]: Give the tight, unqualified upper bound on the runtime (Big-O) of `increment` on a list with n elements in it, and briefly explain your answer.

Question 6 [*3 points*]: Give the tight, unqualified upper bound on the runtime (Big-O) of the following code snippet, and briefly explain your answer. Assume that `node` is an arbitrary node of `list`.

```java
for(int i = 0; i < n; i++) {
    node = increment(list, node)
}
```

Question 7 [*4 points*]: Does your answer to Question 5 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 8 [*5 points*]: Does your answer to Question 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.

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.

Answers given without valid work will receive no credit.

Question 9 [*5 points*]: Let $f(n) = 17n^2 + 5n + 6n\log(n)$. Prove that $f(n) \in O(n^2)$.

Question 10 [*5 points*]: Let $g(n) = 42 + 2^{\log(n^4)} + 6n^3$. Prove that $g(n) \in \Omega(n^4)$.

Question 11 [*10 points*]: Let $h(n) = \log(n^7) + 5n^3 + 14n\log(n)$. Prove that $h(n) \in \Theta(n^3)$

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) $\sum_{i=j}^{k}(f(i) + g(i)) = \sum_{i=j}^{k} f(i) + \sum_{i=j}^{k} g(i)$

(S4) $\sum_{i=j}^{k} f(i) = \sum_{i=l}^{k} f(i) - \sum_{i=l}^{j-1} f(i)$ for any $l < j$

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

(S6) $\sum_{i=j}^{k} f(i) = f(j) + ... + f(l-1) + \sum_{i=l}^{k} f(i)$ for any $j < l \leq k$

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

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

(S9) $\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$

Log Rules

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

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

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

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

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