CS 240: Data Structures and Data Management **Winter 2023**

Midterm Practice Problems

Note: This is a sample of problems designed to help prepare for the midterm exam. These problems do *not* encompass the entire coverage of the exam, and should not be used as a reference for its content.

1 True/False

Indicate "True" or "False" for each of the statements below.

a) If $T_1(n) \in \Omega(f(n))$ and $T_2 \in O(g(n))$, then $\frac{T_1(n)}{T_2(n)} \in \Omega\left(\frac{f(n)}{g(n)}\right)$.

b) If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = e^{42}$, then $f(n) \in \Theta(g(n))$.

c) If $f(n) \in o(n \log n)$, then $f(n) \in O(n)$.

d) All heaps satisfy the AVL height-balance requirement.

e) A binary search tree with n leaves must have height in $O(n)$.

f) If at least one rotation was performed during AVL-delete, then the height of the AVL Tree after deletion is strictly less than the height of the AVL Tree before deletion.

g) A skip-list with n keys and height h must have a total of $\Theta(nh)$ nodes.

h) Given an array of n elements, we can construct a skip-list of the keys in A in $O(n)$ time.

i) If we perform an odd number of search operations in a linked list with the transpose heuristic, the resulting linked list will always be different from the initial linked list.

j) The probability of a skip list tower having height exactly 1 is $\frac{1}{4}$.

2 Order Notation

a) Show that $3n^2 - 8n + 2 \in \Theta(n^2)$ from first principles.

b) Complete the statement $n! \in \sqcup(n^n)$ by filling in $\sqcup$ with either Θ, o, or ω, and prove the corresponding relationship.

c) Prove or disprove: if $f(n) \in \Theta(g(n))$, then $\log f(n) \in \Theta(\log g(n))$. Assume that $f(n)$ and $g(n)$ are positive functions. You should prove the statement from first principles or provide a counter example.

3 Runtime Cases

After midterm grading is complete, Prashanth wants to make sure all the grades are okay, and have them listed in sorted order. He asks Yundi or Deven to get help with this task. Yundi and Deven have many sorting algorithms to choose from, so their decision is based on their mood. Their moods fluctuate while they check assignment grades, and eventually depends on the final student's grade.

Assignment grades in array A are integers ranging from 0 to 100. There is exactly one student with a grade of 0, and exactly one student with a grade of 100. The function `verify(A)` runs in $\Theta(n)$ time and returns:

- +1, if the final student of A scored 100.
- -1, if the final student of A scored 0.
- 0, otherwise.

```
YundiSort (A):
    mood := verify(A)
    if mood = +1
        MergeSort(A)
    else if mood = -1
        SelectionSort(A)
    else MSDRadixSort(A, R = 10, m = 3)
```

```
DevenSort (A):
    mood := verify(A)
    if mood = +1
        SelectionSort(A)
    else if mood = -1
        LSDRadixSort(A, R = 10, m = 3)
    else MergeSort(A)
```

a) What is the worst-case, and average-case runtime for `YundiSort`?

b) What is the worst-case, and average-case runtime for `DevenSort`?

c) Prashanth's decision, which we refer to as `PrashanthSort`, is to flip a coin. If it flips heads, he gets help from Yundi (`YundiSort`). Otherwise, if it flips tails, he gets help from Deven (`DevenSort`). What is the expected runtime (worst-case expected) for `PrashanthSort`?

4 Pseudocode Runtime Analysis

Analyze the worst-case runtimes for the following pseudocodes as functions of n. A $\Theta()$ bound is sufficient.

a)

```
1 j ← 0;
2 k ← 1;
3 while j ≤ n do
4 |   j ← j + k;
5 |   k ← k + 2;
6 end
```

b)

```
1 x ← n;
2 while x > 1 and x < n^12 do
3 |   if x is even then
4 |   |   x ← x/2;
5 |   end
6 |   x ← 3x + 1;
7 end
```

5 Numbers in Range

We have an array $\mathcal{A}$ of n non-negative integers such that each integer is less than k. Give an algorithm with $O(n+k)$ preprocessing time such that queries of the form "how many integers are there in $\mathcal{A}$ that are in the range $[a, b]$?" can be answered in $O(1)$ time.
Note that a and b are not fixed; they are parameters given to the query algorithm.

6 Updating Partial Sum

Consider the problem where we have a sequence of n elements: $S = a_1, a_2, ..., a_n$, and 3 operations:

- $Add(S, b) \rightarrow a_1, a_2, ..., a_n, b$
- $Update(S, i, \Delta) \rightarrow a_1, ..., a_{i-1}, \Delta, a_{i+1}, ..., a_n$
- $PartialSum(S, k) \rightarrow \sum_{i=1}^{k} a_i$

Design a data structure that can perform each of these operations in $O(\log n)$ expected time.

7 MTF Scenario Analysis

Consider a linked list of n items where we perform m searches using the Move-To-Front heuristic, where $m > n$. For each of the following scenarios, give $\Theta()$ bounds on the worst-case runtimes in terms of m and n.

a) 99% of the operations are on the same key x.

b) At most $\sqrt{n}$ different elements are searched.

c) After an element is searched, it is searched again in the next 100 queries, or it is never searched again.

8 Union Selection

Suppose you have two sorted arrays, A and B of sizes m and n respectively, where the elements in the union of A and B are distinct. Given an integer k with $k \leq m + n$, design an $O(\log k)$ algorithm to find the k-th smallest element in the union of A and B.