**CS 240: Data Structures and Data Management** **Spring 2022**

# 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 and an arbitrarily long sequence of coin flips, 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) Show that $\frac{n^2}{n+\log n} \in \Theta(n)$ from first principles.

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

d) Complete the statement $\frac{n^2}{n+\log n} \in \sqcup(n)$ by filling in $\sqcup$ with either $\Theta$, $o$, or $\omega$, and prove the corresponding relationship.

e) 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 Zahra to help with this task. Yundi and Zahra 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)
```

```
ZahraSort (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 `ZahraSort`?

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 Zahra (`ZahraSort`). 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)

```
j ← 0;
k ← 1;
while j ≤ n do
    j ← j + k;
    k ← k + 2;
end
```

b) For the following algorithm, you may assume that $n$ is a power of 3.

**Algorithm 1:** STOOGE$(A, i, j)$

**Input:** Array $A$ of size $n$, index $i$ (initially 0), index $j$ (initially $n-1$)
**Output:** No output but the subarray $A[i \dots j]$ will be sorted

```
if A[j] < A[i] then
    SWAP(A[i], A[j])
end
if j − i + 1 > 2 then
    t ← ⌊(j−i+1)/3⌋;
    STOOGE(A, i, j − t);
    STOOGE(A, i + t, j);
    STOOGE(A, i, j − t);
end
```

c)

```
x ← n;
while x > 1 and x < n^12 do
    if x is even then
        x ← x/2;
    end
    x ← 3x + 1;
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 $d$-ary Heaps

Suppose instead of binary heaps, we have $d$-ary heaps, where each internal node contains $d$ children, except possibly the last one.

a) What is the height of a $d$-ary heap of $n$ nodes?

b) Suppose the $d$-ary heap is represented in an array similar to binary heaps. For a node at index $i$, give the indices of its parent and all of its children.

c) Give an efficient algorithm for `Insert` and analyze its runtime.

d) Given an efficient algorithm for `deleteMax` and analyze its runtime.

## 8 AVL Deletion

In this problem, we will show that deleting a single node in an AVL tree of height h might require $\Theta(h)$ rotations. First, we define a family $T_h$ (for $h \geq -1$) recursively in the following manner: $T_{-1}$ is empty and $T_0$ is a single node. To form $T_h$, we start with a single node and take a copy of $T_{h-2}$ and a copy of $T_{h-1}$ as the left and the right children of the root, respectively.

a) For $h \geq 0$, what is the height of $T_h$? Prove your claim.

b) Prove that for $h \geq 0$, $T_h$ satisfies the height requirements of an AVL tree.

c) On $T_3$, what are the leaves which require $\lfloor 3/2 \rfloor = 1$ rotation upon deletion? Pick one and show the resulting tree.

d) Same question with $T_4$, but now with $\lfloor 4/2 \rfloor = 2$ rotations.

e) Prove by induction that the above construction of $T_h$ results in trees for which there is a node that requires $\lfloor h/2 \rfloor$ rotations upon deletion.

## 9 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.