CMSC 351 Midterm II Solutions Spring 2019

Name (PRINTED): ______________________________

Student ID #: ______________________________

This exam is closed-book, closed-notes, and closed-calculators, except you may use the handout. Show your work. *Clarity and neatness count.*

If you need more space, you can use the blank space in the back. Make sure to cross reference it.

There are six questions. Good luck.

1 (10):
2 (15):
3 (15):
4 (15):
5 (20):
6 (25):

1. (10 points) Professor Emria uses the following algorithm for merging k sorted lists, each having n/k elements. She takes the first list and merges it with the second list using a linear-time algorithm for merging two sorted lists, such as merging algorithm used in merge sort. Then, she merges the resulting list of $2n/k$ elements with the third list, merges the list of $3n/k$ elements that results with the fourth list, and so forth, until she ends up with a single sorted list of all n elements. What is the total number of comparisons in terms of n and k? Show your work.
Note: In this algorithm, it takes n comparisons to merge two sorted lists of combined length of n, rather than $n-1$ comparisons like we saw in merge sort.

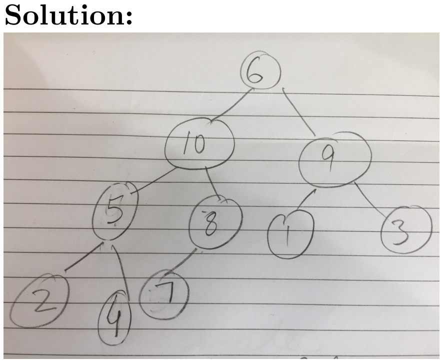
Solution:

$$\begin{aligned} T &= \frac{2n}{k} + \frac{3n}{k} + \ldots + \frac{kn}{k} \\ &= \sum_{i=2}^{k} \frac{in}{k} \\ &= \frac{n}{k} \sum_{i=2}^{k} i \\ &= \frac{n}{k} \frac{(k+2)(k-1)}{2} \end{aligned}$$

2. (15 points) We are in the process of building a max-heap and the array at this instance is given by, $A = [6, 10, 9, 5, 8, 1, 3, 2, 4, 7]$

   (a) Show the heap for the array, A.

   Solution:

   (b) Which node in the heap is in violation of the max heap property?

   Solution:
   root node, with a value of 6.

   (c) How many comparisons are required to convert this heap into a max heap, assuming you know which node is in violation of the max-heap property?

   Solution:
   5

   (d) Show the array for the max-heap after the element is sifted to its correct position and no longer violates the max-heap property.

Solution:

$[10, 8, 9, 5, 7, 1, 3, 2, 4, 6]$

3. (15 points) Find the lower and upper bounds of the following summations:

(a) $\sum\limits_{i=0}^{n-1} \frac{n}{n-i}$

Solution:

$$\sum_{i=0}^{n-1} \frac{n}{n-i} = \frac{n}{n} + \frac{n}{n-1} + \frac{n}{n-2} + \dots + n$$

$$= n\left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}\right]$$

$$= n \sum_{k=1}^{n} \frac{1}{k}$$

$$\int_{k=1}^{n+1} \frac{1}{x}\,dx \leq \sum_{k=1}^{n} \frac{1}{k} \leq 1 + \int_{1}^{n} \frac{1}{x}\,dx$$

$$\ln x \Big|_{1}^{n+1} \leq \sum_{k=1}^{n} \frac{1}{k} \leq 1 + \ln x \Big|_{1}^{n}$$

$$\ln(n+1) \leq \sum_{k=1}^{n} \frac{1}{k} \leq 1 + \ln n$$

Therefore,

$$n \ln(n+1) \leq \sum_{i=0}^{n-1} \frac{n}{n-i} \leq n(1 + \ln n)$$

(b) $\sum\limits_{k=1}^{n} k^2 lg k$

Solution:

$$\sum_{k=1}^{n} k^2 \lg k =$$

$$\int x^2 \lg x\,dx = \frac{1}{\ln 2}\int x^2 \ln x\,dx$$

$$= \frac{1}{\ln 2}\left[\ln x \cdot \int x^2\,dx - \int \frac{d}{dx}\ln x \int x^2\,dx\,dx\right]$$

$$= \frac{1}{\ln 2}\left[\frac{x^3}{3}\cdot \ln x - \int \frac{1}{x}\cdot\frac{x^3}{3}\,dx\right]$$

$$= \frac{1}{\ln 2}\left[\frac{x^3 \ln x}{3} - \int \frac{x^2}{3}\,dx\right]$$

$$= \frac{1}{\ln 2}\left[\frac{x^3 \ln x}{3} - \frac{x^3}{9}\right]$$

For lower bound we have to split the summation

$$\sum_{k=1}^{n} k^2 \lg k \geq 1^2 \lg 0 + \sum_{k=2}^{n} k^2 \lg k$$

$$\geq \int_1^n x^2 \lg x\,dx$$

$$\int_1^n x^2 \lg x\,dx \leq \sum_{k=1}^{n} k^2 \lg k \leq \int_1^{n+1} x^2 \lg x\,dx$$

$$\frac{1}{\ln 2}\left[\frac{x^3 \ln x}{3} - \frac{x^3}{9}\right]\Bigg|_1^n \leq \sum_{k=1}^{n} k^2 \lg k \leq \frac{1}{\ln 2}\left[\frac{x^3 \ln x}{3} - \frac{x^3}{9}\right]\Bigg|_1^{n+1}$$

$$\frac{1}{\ln 2}\left[\frac{n^3 \ln n}{3} - \frac{n^3}{9} + \frac{1}{9}\right] \leq \sum_{k=1}^{n} k^2 \lg k \leq \frac{1}{\ln 2}\left[\frac{(n+1)^3 \ln(n+1)}{3} - \frac{(n+1)^3}{9} + \frac{1}{9}\right]$$

4. (15 points) Solve the following recurrence equations:

(a) $T(n) = T\left(\frac{n}{7}\right) + T\left(\frac{6n}{7}\right) + n, \quad T(1) = 0$

Solve for an upper bound using constructive induction.

Solution:

$$T(n) = T\left(\frac{n}{7}\right) + T\left(\frac{6n}{7}\right) + n$$

Guess: $T(n) \le a n \lg_7 n$

Base Case: $n = 1$: $T(1) = a \cdot 1 \cdot \lg_7 1 = 0$

I.H.: Holds for all $\le n-1$

Inductive Step: Show for n

$$T(n) = \frac{an}{7} \lg_7 \frac{n}{7} + \frac{6na}{7} \lg_7 \frac{6n}{7} + n$$

$$= \frac{an}{7} \lg_7 n - \frac{an}{7} + \frac{6an}{7} \lg_7 n + \frac{6an}{7} \lg_7 6 - \frac{6an}{7} + n$$

$$= an \lg_7 n - an + \frac{6an}{7} \lg_7 6 + n$$

$$= an \lg_7 n + \left(-a + \frac{6a}{7} \lg_7 6 + 1\right) n$$

$$\le an \lg_7 n$$

$$-a + \frac{6a}{7} \lg_7 6 + 1 \le 0$$

$$a\left(-1 + \frac{6}{7} \lg_7 6\right) \le -1$$

$$a\left(1 - \frac{6}{7} \lg_7 6\right) \ge 1$$

$$a \ge \frac{7}{7 - 6 \lg_7 6} \Rightarrow T(n) \le \frac{7}{7 - 6\lg_7 6} n \lg_7 n$$

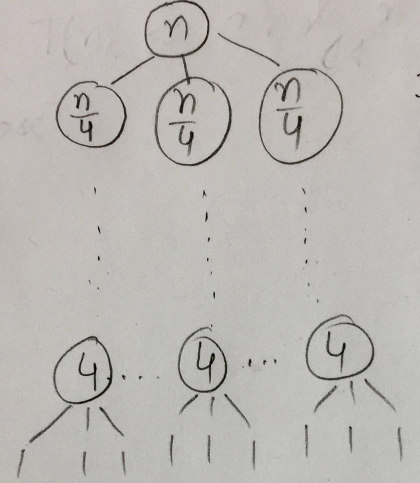
$$T(1) = 0$$

(b) $T(n) = 3T\left(\frac{n}{4}\right) + n^2, \quad T(1) = 1$

Solve using the recursion tree.

Solution:

$$T(n) = 3\,T\left(\frac{n}{4}\right) + n^2, \quad T(1) = 1$$



n^2

$3 \cdot \left(\frac{n}{4}\right)^2$

$3^{\lg_4 n - 1}\left(\frac{n}{4^{\lg_4 n - 1}}\right)^2$

$n^{\lg_4 3} \cdot 1$

$$T(n) = \sum_{i=0}^{\lg_4 n - 1} 3^i \left(\frac{n}{4^i}\right)^2 + n^{\lg_4 3}$$

$$= n^2 \sum_{i=0}^{\lg_4 n - 1} \left(\frac{3}{16}\right)^i + n^{\lg_4 3}$$

$$= n^2 \left[\frac{1 - \left(3/16\right)^{\lg_4 n}}{1 - 3/16}\right] + n^{\lg_4 3}$$

$$= n^2 \cdot \frac{16}{13}\left[1 - \frac{3^{\lg_4 n}}{16^{\lg_4 n}}\right] + n^{\lg_4 3}$$

$$= n^2 \cdot \frac{16}{13}\left[1 - \frac{n^{\lg_4 3}}{n^2}\right] + n^{\lg_4 3}$$

$$T(n) = \frac{16}{13}n^2 - \frac{16}{13}n^{\lg_4 3} + n^{\lg_4 3} = \frac{16}{13}n^2 - \frac{3}{13}n^{\lg_4 3}$$

$$T(1) = 16/13 - 3/13 = 1$$

5. (20 points) Suppose you are choosing between the following three algorithms:

(a) Algorithm X solves problems of size n by dividing them into four sub-problems of half the size, recursively, solving each sub-problem. The work done between the recursive calls is n^2. Assume T(1) = 1.

- What is the recurrence equation?

> **Solution:**
> $T(n) = 4T\left(\frac{n}{2}\right) + n^2, \qquad T(1) = 1$

- What is the runtime of the algorithm?

> **Solution:** $O(n^2 lgn)$

(b) Algorithm Y solves problems of size n by recursively solving two sub-problems of size $n-1$ and the work done between the recursive calls is constant with a value of 5. Assume T(0) = 1.

- What is the recurrence equation?

> **Solution:**
> $T(n) = 2T(n-1) + 5, \qquad T(0) = 1$

- What is the runtime of the algorithm?

> **Solution:** $O(2^n)$

(c) Algorithm Z solves problems of size n by dividing them into nine sub-problems of size $n/3$, recursively solving each sub-problem. The work done between recursive calls is n. Assume, T(1) = 1.

- What is the recurrence equation?

> **Solution:**
> $T(n) = 9T\left(\frac{n}{3}\right) + n, \qquad T(1) = 1$

- What is the runtime of the algorithm?

> **Solution:** $O(n^2)$

(d) Which algorithm would you choose?

> **Solution:** Algorithm Z, since that is the fastest.

6. (25 points) You are given two sequences of numbers A and B, of lengths n, each, and a real value, S. Find two elements, one each from A and B, whose sum equals S. Assume there will be at least 1 element in each of the arrays.

   (a) Write pseudocode for an efficient brute-force algorithm. What is its runtime in the worst-case?

   Solution:

   ```
   for i=1 to n:
       for j= 1 to n:
           if A[i] + B[j] == S:
               return i,j
   return -1,-1
   ```

   (b) Write pseudocode for an optimal solution to find the two elements. What is its runtime in the worst-case?

   Solution:

   ```
   Mergesort A
   for j= 1 to n:
       d = S - B[i]
       i = BinarySearch(A,d,1,n)
       if(i != -1):
           return i,j
   return -1,-1

   BinarySearch(A,d,start,end):
       if start > end:
           return -1
       else:
           mid = (start+end)/2
           if A[mid] == d:
               return mid
           elif A[mid] < d:
               return BinarySearch(A,d,mid+1,end)
           else:
               return BinarySearch(A,d,start,mid-1)
   ```