EECS4020 Algorithms

Exam 1

Time: 10:10am – 12:00pm, March 30, 2022

Answer All Questions. Total Marks = 105. Maximum Score = 105/100.

Important: (1) When you design an algorithm, you need to show its correctness, and analyze its running time. (2) When you get stuck, try another question first.

1. (15%)

   (a) Let $A[1..n]$ be a sorted array of distinct numbers. Your friend, Peter, told you that these n numbers are exactly chosen from $1, 2, \ldots, n, n+1$, but with one of them missing.

   (10%) Show how to find the missing number in $O(\log n)$ time.

   (b) When you are about to find the missing number, you decide to take a break first. During the break, your mischievous friend, John, has divided your array into two parts $A_{\mathsf{left}} = A[1..i]$ and $A_{\mathsf{right}} = A[i+1..n]$, and re-arrange the array so that he puts A_{right} in front of A_{left}; precisely, the array now becomes $A[i+1..n]A[1..i]$. See Figure 1 for an example.

   (5%) Show how to find the missing number in $O(\log n)$ time.

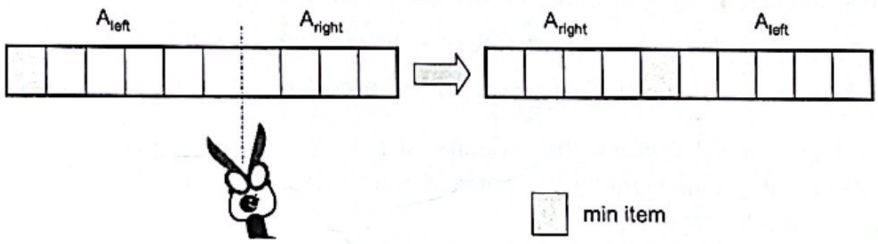

Figure 1: John's modification to the array.

2. Suppose that

$$T(n) = T(n/3) + T(n/4) + T(5n/12) + n$$

   (15%) Show that $T(n) = \Theta(n \log n)$.

3. Consider performing the following sequence of 10 operations on an empty binary min-heap:

   Insert 5, Insert 7, Insert 6, Insert 1, Insert 3,
   Insert 4, Extract-Min, Insert 8, Insert 2, Extract-Min

   (15%) Show the resulting heap after the last operation.

   Note: Binary min-heap is the heap that we introduced in the lecture.

4. The following is a pseudo-code for `BubbleSort` on an array A:

```
BubbleSort(A)
for phase k = 1, 2, ..., n − 1
    for position j = 1, 2, ..., n − 1
        if A[j] ≥ A[j + 1]
        { Swap the entries A[j] and A[j + 1]; }
```

(10%) Explain why it is not stable, and make a minor change to make it stable.

5. (a) (5%) Show that why, for any comparison sort algorithm, sorting four distinct numbers requires at least 5 comparisons in the worst case.

   (b) (10%) Give a comparison sort algorithm that sorts any four distinct numbers with at most 5 comparisons.

6. (15%) Show the steps of how to use RADIXSORT to sort the following list of English words:

   BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX,

   COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB.

7. Consider a set of $3n$ distinct numbers. Let M be its median, whose value is not known. It is known that these $3n$ numbers are distributed in three arrays $A[1..n]$, $B[1..n]$, and $C[1..n]$. Inside each array, the numbers are sorted in increasing order.

   (10%) Show how to check if M lies in array A or not, in $O(\log n)$ time.

   Remark: You may assume that $3n$ is an odd number, so that the median is well-defined.

8. We have a bag which has a weight capacity of x kilograms. We also have n items, each of different weights. Our target is to pack as many items as possible in the bag, without exceeding the weight capacity.†

   Your clever friend, Matthew, tells you that one way is to pack items according to increasing order of the weights, from smallest to the largest (until exceeding the weight capacity), and this strategy can pack the maximum number of items.

   You want to follow Matthew's idea. From the first glance, it seems that finding all the desired items to be packed will require sorting, thus taking $\Theta(n \log n)$ time. However, this can be done much faster.

   (10%) Design an algorithm to determine the desired items in $O(n)$ time.

   Note: You may assume that Matthew's idea is correct, so that you don't need to show correctness. However, you still need to anaylze the running time.

† Note that we want to maximize the number of items, not the total weights of the items.