D&C, Recurrences, Greedy Algorithms

Instructor: Piyush Kumar *Problem Set 2*

Write in your own words, give proofs of correctness. You MUST submit your own copy written in your own words. Discuss the problems with your partner and then write the answers in your own words independently. This homework will be due BEFORE class on the date posted on the website. Late submissions will not be accepted.

Exercise 1

1. Given an array storing numbers ordered by value, modify the binary search routine to return the position of the first number with value x in the situation where x can appear multiple times in the array. Be sure that your algorithm is $\Theta(\log n)$, that is, do not resort to sequential search once an occurrence of x is found.

2. Suppose you are given an array A of n sorted numbers that has been circularly shifted k positions to the right. For example, $\{35, 42, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted $k = 2$ positions, while $\{27, 29, 35, 42, 5, 15\}$ has been shifted $k = 4$ positions.

   (a) Suppose you know what k is. Give an $O(1)$ algorithm to find the largest number in A.

   (b) Suppose you do not know what k is. Give an $O(\log n)$ algorithm to find the largest number in A. For partial credit, you may give an $O(n)$ algorithm.

3. A max-min algorithm finds both the largest and smallest elements in an array of n values. Design and analyze a divide-conquer max-min algorithm that uses $(\lceil 3n/2 \rceil) - 2$ comparisons for any integer n. (Hint: first consider the case where n is a power of 2.)

4. Give an efficient divide-and-conquer algorithm to find the kth largest element in the merge of two sorted sequences S1 and S2 . The best algorithm runs in time $O(\log(max(m, n)))$, where $|S_1| = n$ and $|S_2| = m$.

5. Given an array of n real numbers, consider the problem of finding the maximum sum in any contiguous subvector of the input. For example, in the array

$$\{31, -41, 59, 26, -53, 58, 97, -93, -23, 84\}$$

   the maximum is achieved by summing the third through seventh elements, where 59 + 26 + (-53) + 58 + 97 = 187. When all numbers are positive, the entire array is the answer, while when all numbers are negative, the empty array maximizes the total at 0.

   (a) Give a simple, clear, and correct $O(n^2)$-time algorithm to find the maximum contiguous subvector.

   (b) Give a correct $O(n \log n)$ divide-and-conquer algorithm for this problem.

Exercise 2

Solve the following recurrences.

1. $T(n) = 2T(n/2) + n/lgn$
2. $T(n) = 4\sqrt{n}T(\sqrt{n}) + n$
3. $T(n) = T(n - \sqrt{n}) + O(1)$

Exercise 3

The natural greedy algorithm for making change of n units using the smallest number of coins is as follows. Give the customer one unit of the highest denomination coin of at most n units, say d units. Now repeat to make change of the remaining $n - d$ units. For each of the following nations coinage, establish whether or not this greedy algorithm always minimizes the number of coins returned in change. If so, prove it, if not give a counter example.

1. United States coinage, which consists of half dollars (50 cents), quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies.
2. English coinage before the decimalization, which consisted of half-crowns (30 pence), florins(24 pence), shillings (12 pence), sixpence (6 pence), threepence (3 pence), pennies (1 pence), half pennies (1/2 pence), and farthings (1/4 pence).
3. Portuguese coinage, which includes coins for 1,2.5,5,10,20,25 and 50 escudos. You need only consider change for an integer number of escudos.
4. Martian coinage, where the available denominations are $1, p, p^2, ..., p^k$ where $p > 1$ and $n \geq 0$ are integers.