ESO 207

Practice Problems - 1

Devendra Agrawal and Rachit Nimavat
TA, ESO207

August 9, 2014

These are practice problems. They may be discussed in classes and/or solutions to the problems will be posted. These problems are not for submission.

1. *General.* Given a positive integer n and a list containing $n-1$ distinct integers in the range $[1, n]$, design an $\mathcal{O}(n)$ time algorithm to find the missing number. You are NOT allowed to modify the list even temporarily. Your algorithm is allowed to use only $\mathcal{O}(1)$ words of extra space. A word is of size $O(\log n)$ bits.

   (a) Similar problem as above, but now suppose you are given $n-2$ distinct integers in the range $[1, n]$. Design an $O(n)$ time algorithm using $O(1)$ words of extra space to find the missing numbers.

   (b) Similar to above, but suppose you are given $n-k$ distinct integers in the range $[1, n]$. Design an $O(n)$ time algorithm using $O(k)$ words of extra space to find the missing numbers.

2. *Lists*

   (a) Write an iterative as well as recursive function in C for reversing a singly linked list. Can you do it using just $\mathcal{O}(1)$ extra space?

   (b) Repeat this exercise for a doubly linked list. Design the structure so that this can be done in $O(1)$ time? Does this solution work for singly linked lists?

   (c) Given a doubly linked list, for each element report its difference with the corresponding element of the reversed for of the same list. You should use $\mathcal{O}(1)$ extra space. Example: For $5- > 4- > 3- > 2- > 1$, the answer would be $4, 2, 0, -2, -4$

3. *Heap/Priority Queue.* Given an array of size n and an integer k, design an algorithm that makes a pass over the array and reports the maximum element in each contiguous subarray (window) of size k. An $O(nk)$ time algorithm is straightforward. Design an $O(n \log k)$ time algorithm. A more efficient algorithm may be possible. If you want to code it, you can try submitting it here.

4. *Stacks.* Given a parenthesis array, design an algorithm to determine whether the array is a valid paranthesis expression or not. That is, determine whether the parenthesis in that expression are correctly balanced and each left bracket precedes its matching right bracket.

5. In an integer array with n elements (n is quite large), find the minimum valued k elements ($k < n$) with complexity smaller than $\mathcal{O}(n \log n)$. Both $O(nk)$ time algorithm (find the minimum, the second minimum, etc.) and an $O(n \log n)$ algorithm (sort and report the minimum k elements) are straightforward. Design an $O(n \log k)$ time algorithm.

6. Convert a decimal number to its binary representation using a stack.

7. *Queues/Breadth-first search.* Consider a supernatural knight k on a normal nxn chessboard. Apart from normal 'Two and Half steps', using his supernatural power, he can also move vertically (and **not** horizontally) any number of steps on the board. Given any two squares on the chessboard, determine minimum number of steps in which k can move from one square to another.