Assignement 1, COSC 3P03, Algorithms, Winter, 2016

Due: 5:00 pm, Jan. 28, Thursday.

1. (10) Modify the mergesort as follows: instead of dividing the input list into two lists, the list is divided into three sublists of (roughly) the same size. (a) Write down the recursive algorithm; (b) Let $t(n)$ be the running time of your algorithm, write a recurrence for $t(n)$; (c) What is the running time and how does it compare to the traditional mergesort? Note that you need to solve for $t(n)$ first. Note: if you want to merge two lists L_1 and L_2, you can just say in your algorithm "Merge L_1 and L_2."

2. (5) Let $A[1..n]$ be a sorted array of distinct integers, some of which may be negative. Give a recursive $O(\log n)$ algorithm that can find an index i such that $1 \leq i \leq n$ and $A[i] = i$, provided that such an index exists. This means that you need to give the algorithm first and then analyze it.

3. (15) Consider the following three algorithms for the problem of computing x^n, $x \neq 0$ and $n \geq 0$. For each algorithm, find its asymptotic running time. Show your work, which means if the algorithm is recursive, you need to give its recurrence (don't forget the initial condition) and solve it in whatever way.

   $A_1(x, n)$
   $r = 1$
   for i=1 to n
       $r = r \times x$
   return r

   $A_2(x, n)$
   if $n = 0$
       return 1
   else
       return $(x \times A_2(x, n-1))$

   $A_3(x, n)$
   if $n = 0$
       return 1
   else
       $t = A_3(x, \lfloor n/2 \rfloor)$
       if n is even
           return $t \times t$
       else
           return $t \times t \times x$

4. (10) List the functions below from lowest order to highest order, where $k = 3/2$ and $c = \log 3$ are constants. If any two or more are of the same asymptotic order, group them together.

$$\log^k n,\ \log n^k,\ n^k,\ k^n,\ 2^n,\ 2^{cn},\ 2^{n+1},\ n^{\log k},\ k^{\log n},\ \log_3^5 n,\ \log_{100}^{500} n,\ \log(n!),\ \log(n^n),$$
$$100n + \log n,\ n + (\log n)^2,\ \log n,\ \log(n^2),\ n^2/\log n,\ n\log^2 n,\ (\log n)^{\log n},\ n/\log n,\ n^{1/2},\ (\log n)^5,\ n2^n,\ 3^n.$$

5. (15) Use as many methods as possible to solve the following recurrence:

$$t(n) = 2t(n/2) + 1.$$

6. (20) Solve the following using whatever methods.

   a. $t(n) = 2t(n/2) + n^3$;

   b. $t(n) = t(n/10) + n$;

   c. $t(n) = 16t(n/4) + n^2$.

   d. $t(n) = t(n-1) + \log n$.

7. (15) Solve the following recurrences by the masters method, if possible. In case(s) where the method does not apply, state why and solve it/them by other means.

$$\begin{aligned} t(n) &= 9t(n/3) + n \\ t(n) &= t(2n/3) + 1 \\ t(n) &= 3t(n/4) + n \log n \end{aligned}$$

8. (10) What is the Big Oh function for each of the following functions of n (it should be as tight and as simple as possible). There is no need to justify your answers.

   a. $t(n) = (n \log n)/2 + f(n)$, where $f(n) = o(n \log(n^{100}))$

   b. $t(n) = 1 + 3 + 5 + ... + (n-1)$, where n is even

   c. $t(n) = \log^2 n + \log(n^3)$

   d. $t(n) = 4^{\log n}$

   e. $t(n) = 1 + 1/2 + 1/2^2 + ... + 1/2^n$

   f. $t(n) = n^1 + n^2 + n^3 + ... + n^k + 2^n$, where $k > 0$ is an integer

   g. $t(n) = 4n^{3/4} + 5n \log n + 2n \log\log n$

   h. $t(n) = 2016 + \sin(n)$

   i. $t(n) = 2t(n-1) + 1$, $t(1) = 1$ (this the number of moves for the Tower of Hanoi)

   j. $t(n) = (n^2 - 1)/(n + 1)$