Basic Algorithms, Problem Set 2

Due by 8 a.m. Wednesdy, February 10.

Send to Jingshuai: jj2903@nyu.edu.

He who learns but does not think is lost. He who thinks but does not learn is in great danger. – Confucius

1. Illustrate the operation of PARTITION(A,1,12) on the array

$$A = (13, 18, 9, 5, 12, 8, 7, 4, 11, 2, 6, 10)$$

(You may use either the text's program or the version given in class, but please specify which you are using.)

2. Let $L(n)$, ("L" for lucky) denote the number of comparisons that quicksort does if each time it is applied the pivot lies in the precise center of the array. For example, applying quicksort to an array of length 31, say $A(1) \cdots A(31)$ objects, there would be 30 comparisons (between $A(31)$ and all the other $A(j)$) and then $A(31)$ would end up in the 16^{th} place and there would be two recursive calls to quicksort on arrays each of size 15. Find the *precise* value of $L(1023)$. (Hint: thats one less than 1024!)

3. Babu[1] is trying to sort a, b, c, d, e with seven comparisons. First he asks "Is $a < b$" and the answer is yes. Now he asks "Is $a < c$?" Argue that (in worst-case) he will not succeed.

4. Illustrate the operation of `COUNTING-SORT` with $k = 6$ on the array $A = (6, 0, 2, 2, 0, 1, 3, 4, 6, 1, 3)$.

5. You are given a Max-Heap with n entries. Assume all entries are distinct. Your goal is to find the *third largest* entry. One way would be to `EXTRACT-MAX` twice and then `MAXIMUM`. How long does this take? Find a better (by which we always mean faster for n large) way.[2]

[1] former student, now big cheese at GE Hyderabad

[2] A major goal of this course is to analyze the asymptotic time for an algorithm. As such, you will sometimes be given algorithms to analyze that are not at all optimal – but they should be analyzed for their own sake.

6. Assume [3] a program NEARMED[A,p,r] which returns an i such that $A[i]$ is uniform between the 49-th and 51-st percentile. (e.g., between $p+0.49(r-p)$ and $p+0.51(r-p)$) when $A[p\cdots r]$ is sorted, and that NEARMED takes constant time K. (By time we mean number of comparisons.) (As a default assume NEARMED produces the median when given less than 100 values.) Create a variant $VQUICK[A,p.r]$ of QUICKSORT that calls on NEARMED to find pivot. Let $VT(n)$ be the expected time for $VQUICK[A,1,n]$. Give (but do *not* attempt to solve!) the recursive equation for $VT(n)$. (Don't worry about initial conditions.) (To clarify: NEARMED simply finds the position i, you still have to appropriately apply PARTITION to get things on its left and right sides.)

   Tell me, what do you plan to do with your one wild and precious life? – Mary Oliver, The Summer Day

[3] While NEARMED is fictitious we get something similar by taking 101 random values, sorting them, and returning their median