Prof. Spencer
March 27, 2017

FUNDAMENTAL ALGORITHMS MIDTERM

> Every true scientist has in his lifetime just one idea, and his life is its embodiment. – Nikolai Luzin

Maximum Score 105. Do all problems.

1. (15) Let $A[1 \cdots N]$ be an array with all entries integers between 0 and $N-1$. How long would `RADIX-SORT` take to sort A assuming that we use base 2 (that is, binary)? (Assume the enries $A[I]$ are already given as binary strings in the input.) You *must* give an argument for your answer. Give (no proofs required!) a *faster* way to sort this data.

2. (15) `Toom-3` is an algorithm similar to the Karatsuba algorithm discussed in class. (Don't worry how `Toom-3` really works, we just want an analysis given the information below.) It multiplies two n digit numbers by making five recursive calls to multiplication of two $n/3$ digit numbers plus thirty additions and subtractions. Each of the additions and subtractions take time $O(n)$. Give the recursion for the time $T(n)$ for `Toom-3` and use the Master Theorem to find the asymptotics of $T(n)$. Compare with the time $\Theta(n^{\log_2 3})$ of Karatsuba. Which is faster when n is large?

3. (20) Let A be an array of length 127 in which the values are distinct and in increasing order.

   (a) In the procedure `BUILD-MAX-HEAP(A)` *precisely* how many times will two elements of the array be exchanged? (Reason, please!)

   (b) Now suppose the values are distinct and in decreasing order. Again, in the procedure `BUILD-MAX-HEAP(A)` *precisely* how many times will two elements of the array be exchanged? (Reason, please!)

4. (20) Give an algorithm `TINYPIECES` that does the following. As *input* you have an array $PRICE[1 \cdots N]$ where, for $1 \leq i \leq N$, $PRICE[i]$ is the price of a rod of length i. You are given a rod of total length N^5. You wish to cut it into pieces (*but* all pieces must be of length at most N) so as to maximize the total price. Your algorithm should output $VALUE$, where this represents the maximal total price. (Note: You are not being asked to find the actual cutting of the rod.) Analyze (in Θ-land) the total time your algorithm takes. You **must** give a description in clear words of what the algorithm is doing.

5. (20) Describe the algorithm `QUICKSORT`(p,r) which sorts the elements $A[i]$, $p \leq i \leq r$. (You can assume $p \leq r$.) You may, and should, use auxilliary arrays. Subroutines must be described in full. Explain *in clear words* what the algorithm is doing. Give (without proof!) both the average and the worst-case time for `QUICKSORT`(1,n).

6. (15) Here is a psuedocode sorting algorithm [1] that uses Binary Search Tree. We wish to sort $A[1 \cdots N]$. (There are no records here, each $A[I]$ is itself the key.) Begin with an empty BST T.
   Part I: FOR $I = 1$ to N; INSERT $A[I]$ into T; ENDFOR
   Part II: Apply IN-ORDER-TREE-WALK to T
   Analyze *both* the average time and the worst case time for this algorithm.

   Comstock grins and says, 'You sound awfully sure of yourself, Waterhouse! I wonder if you can get me to feel that same level of confidence.'
   Waterhouse frowns at the coffee mug. 'Well, it's all in the math,' he says, 'If the math works, why then you *should* be sure of yourself. That's the whole point of math.'
   from *Cryptonomicon* by Neal Stephenson

[1] thanks to Ben Cullaj for the idea