COMP3804 Winter 2006 — Assignment 2 — Due Thursday Feb. 16

This assignment is due in class on Thursday February 16th. If you are not able to come to class then please put your assignment in the COMP3804 assignment box in the CCSS lounge. The assignments will be picked up from this box immediately after class.

1 Sorting the First k Elements

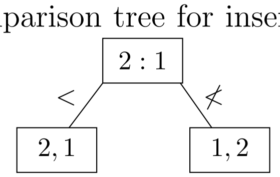
Give an $O(n + k \log k)$ time algorithm for sorting the k smallest elements of an array $A_1, \ldots, A_n$. (Hint: A heap on n elements can be constructed in $O(n)$ time and a heap on $O(k)$ elements supports insertions and deletions in $O(\log k)$ time.)

2 The Insertion-Sort Comparison Tree

Here is the insertion sort algorithm for sorting an array $A_1, \ldots, A_n$ of n elements:

for $i = 2, \ldots, n$ **do**
  $j \leftarrow i$
  while $j > 1$ and $A_j < A_{j-1}$ **do**
    swap $A_j \leftrightarrow A_{j-1}$
    $j \leftarrow j - 1$

1. Draw the comparison tree for insertion sort on arrays of size 3. Label each internal node with the indices of two array indices being compared and label each left with a permutation of $\{1, 2, 3\}$ that results in the computation terminating at that leaf. (Hint: Here is a picture of the comparison tree for insertion sort on arrays of size 2.)



2. What is the length of the longest root-to-leaf path in your picture?

3. In the comparison tree for insertion sort on arrays of size n, what is the exact length of the longest root to leaf path?

3 Sorting Words

Suppose you are given a list of n english words having at most 8 letters each. Describe an $O(n)$ time algorithm to sort this list of words in lexicographic (dictionary) order.

4 Median Finding Algorithms

Analyze the worst-case running times of the following algorithms for finding the element of rank i in an array of n distinct values:

1. Pick three elements from the input, compute the median of these three elements and use this as the pivot.
2. Partition the elements into $n/3$ groups each of size 3. Compute the median of each group and use the median of medians as a pivot.
3. Partition the elements into $n/7$ groups each of size 7. Compute the median of each group and use the median of medians as a pivot.

5 Finding a Weighted Median

Suppose you are given a list of pairs of positive numbers (a_i, w_i), $1 \leq i \leq n$ and let $W = \sum_{i=1}^{n} w_i$. The *weighted median* of S is the value a_i such that

$$\sum_{a_j < a_i} w_j < W/2 \text{ and } \sum_{a_j > a_i} w_j \leq W/2 \ .$$

Describe and analyze an $O(n)$ time algorithm for finding the weighted median. (Hint: You may use the $O(n)$ time median-finding algorithm discussed in class as a subroutine.)

6 A Refined Analysis of Karatsuba's Algorithm

In class we saw Karatsuba's Algorithm, which allows us to multiply two n-digit numbers using $O(n^{\log_2 3})$ arithmetic operations on numbers having $O(1)$ digits. Explain how Karatsuba's algorithm would work for two numbers, one having n digits and one having m digits ($n \geq m$). Show, by giving and solving a recurrence, that the running time of this algorithm is $O(nm^{\log_2 3-1})$.

7 Strassen's Algorithm for Non Powers of 2

In class we saw Strassen's Algorithm for multiplying two $n \times n$ matrices in $O(n^{2.81})$ time, when n is a power of 2. Show how to use this algorithm to multiply an two $m \times m$ matrices in $O(m^{2.81})$ time, even when m is not a power of 2.