CMPS 2200 Introduction to Algorithms – Fall 12

10/1/12

# 4. Homework

Due **10/9/12** in the lab

1. **Best case for quicksort (5 points)**
   Let "Deterministic Quicksort" be the non-randomized Quicksort which takes the first element as a pivot, using the partition routine that we covered in class on slide 4 of the quicksort slides.

   In the best case the pivot always splits the array in half, for all recursive calls of Deterministic Quicksort. Give a sequence of 3 distinct numbers, a sequence of 7 distinct numbers, and a sequence of 15 distinct numbers that cause this best-case behavior. (For the sequence of 15 numbers the first two recursive calls should be on sequences of 7 numbers each, and the next recursive calls on sequences of 3 numbers).

2. **Decision tree (5 points)**
   Below is the code for *Bubble Sort*:

```
void bubbleSort(int A[1..n]){
  for(int i=1; i <= n; i++)
    for(int j=n; j >= i+1; j--)
      if(A[j]<A[j-1])
        swap(A[j],A[j-1]);

}
```

   Draw the decision tree for Bubble Sort for an array $A[1..3]$ of $n = 3$ elements. Annotate the decision tree with comments indicating the part of the algorithm that a comparison belongs to.

3. **Lower bound for comparison-based searching (5 points)**

   Consider the problem of searching for a given key in a sorted array of $n$ numbers. Use a decision tree to show a lower bound of $\Omega(\log n)$ for any comparison-based search algorithm. *(Hint: The decision tree needs to represent the output of the search algorithm in its leaves. What should be stored in the leaves?)*

4. **Close to the median (5 points)**

   Let $A[1..n]$ be an unsorted array of $n$ numbers, and let $k$ be a positive integer $k \leq n$. Assume both $n$ and $k$ are odd. The task is to design an algorithm that outputs the $k$ numbers that are the closest to the median. These are the numbers whose $k$ *rank* is closest to $n/2$. For example, if $A = [2, 1, 7, 4, 6, 5, 8]$ and $k = 3$ then the output should be the numbers $5, 4, 6$. Note that the numbers do not necessarily have to be output in sorted order.)

   Give pseudocode for such an algorithm that runs in $O(n)$ time (either expected, or worst-case). *(Hint: Use the Select algorithm.)*

Flip over to back page $\Longrightarrow$

# Practice Problems

**(Not required for homework credit.)**

1. **Quicksort decision tree**

   The decision tree on the slides in class was for insertion sort of three distinct elements. Draw the decision tree for quicksort of an array $A[1..3]$ of three elements. Use the deterministic quicksort and the partition routine described on the slides. Annotate the decision tree with the current state of the array.

2. **Median computation**
   Suppose arrays $A$ and $B$ are **both sorted** and contain $n$ elements each. Give a divide-and-conquer algorithm to find the median of $A \cup B$ in $O(\log n)$ time. (Describe it either in words or as pseudo-code; whatever you prefer). Argue **shortly** why the runtime is $O(\log n)$.