CS 3343 Analysis of Algorithms – Fall 11

10/4/11

6. Homework

Due: Tuesday 10/18/11 before class

Justify all your answers.

1. **Decision tree (6 points)**

   Draw **the right half** of the decision tree for Insertion Sort for an array $A[1..4]$ of $n = 4$ elements. Annotate the decision tree with comments as to which algorithm part a comparison belongs to.

2. **Radix sort variants (6 points)**

   (a) (2 points) Assume an implementation of radix sort that uses merge sort, instead of counting sort, as the auxiliary stable sort algorithm. When sorting n computer words of b bits, and breaking each word into b/r base-2^r digits what is the runtime of radix sort and what is the optimal choice of r to achieve it?

   (b) (4 points) Suppose you want to lexicographically sort n strings, which each have length at most l. Give pseudo-code that implements a variant of radix sort to lexicographically sort these strings. What is the runtime of your algorithm? *(You may assume that an auxiliary stable sorting algorithm is given; you do not have to provide pseudo-code for it.)*

3. **Speeding up deterministic quicksort (4 points)**

   Consider deterministic quicksort.

   (a) (1 point) Describe what properties the pivots have to have in order to achieve the best-case behavior of deterministic quicksort.

   (b) (3 points) Modify the partition routine by using the SELECT algorithm to choose a "good" pivot. Show how this modification results in a worst-case runtime of $O(n \log n)$ for deterministic quicksort.

Flip over to back page $\Longrightarrow$

Practice Problems

(Not required for homework credit.)

1. **Decision tree (6 points)**

   Draw **the left half** of the decision tree for Insertion Sort for an array $A[1..4]$ of $n = 4$ elements. Annotate the decision tree with comments as to which algorithm part a comparison belongs to.

2. **Radix sort**

   (a) What is the relation between the k in the runtime of counting sort and the b in the runtime of radix sort? You should be able to give a formula that involves both b and k.

   (b) Express 173 as a binary number (base-2), as a base-4 number, and as a base-16 number.

3. **Median computation**
   Suppose arrays A and B are **both sorted** and contain n elements each. The task is to develop a (deterministic) divide-and-conquer algorithm to find the median of $A \cup B$ in $O(\log n)$ time.

   (a) Assume your divide-and-conquer approach is to split the problem into halves. What exactly does that mean in this case?

   (b) In order to arrive at an algorithm with $O(\log n)$ runtime, what runtime recurrence would the algorithm have? This should give you an idea on (i) how many recursive calls you can have and (ii) how much time you have for the dividing and combining steps of the algorithm.

   (c) In the combine step you should be able to discard large parts of the input arrays. Argue why. *(Hint: Make up examples with real numbers in order to better understand what is going on.)*

   (d) Describe the algorithm in words. Shortly argue why the runtime is $O(\log n)$.