CS 5633 Analysis of Algorithms – Spring 10

2/4/10

4. Homework

Due **2/11/10** before class

1. **Mergesort decision tree (5 points)**
   The decision tree on the slides in class was for insertion sort of three elements. Draw the decision tree for mergesort of an array $A[1..3]$ of three elements. Note that the first split is $A[1,2]$ and $A[3]$. Annotate the decision tree with the current state of the array.

   comments about the location in the pseudocode.

2. **Sorting (6 points)**
   Given n numbers between 0 and $f(n)-1$. For each of the functions $f(n)$ below give the runtimes for a) mergesort, b) counting sort, c) for radix sort, and indicate the asymptotic ranking of these different sorting algorithms.

   i) $f(n) = n^3$ ii) $f(n) = 2^n$ iii) $f(n) = n!$

3. **Radix sort with most significant digit first (4 points)**
   Try to sort the numbers

   $$424, 178, 444, 433, 119, 115, 146, 346, 345, 326, 174$$

   using radix sort but starting with the **most** significant digit (so, from left to right, not from right to left).

   Why would a program that implements this strategy be much more complicated than the radix sort that starts with the least significant digit? (Hint: What kind of variables or data structures would you have to maintain?)

4. **Deterministic select (5 points)**
   Consider the following variations of the deterministic selection algorithm:

   a) Divide into groups of 7. Show that the runtime proof of $O(n)$ still works in this case.

   b) Divide into groups of 3. Show where you run into problems when trying to prove a runtime of $O(n)$.

5. **Median computation (6 points)**
   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 $O(\log n)$. *Hint: Take a look at the select algorithm.*

FLIP OVER TO BACK PAGE ⟹

Related questions from previous PhD Exams

Just for your information. You **do not** need to solve them for homework credit.

P1 (a) A comparison sort is an algorithm that sorts based only on comparisons between pairs of input elements. Which of the following sorting algorithms are comparison sorts: bucket sort, counting sort, heapsort, insertion sort, merge sort, quicksort, radix sort?

(b) A binary decision tree is a binary tree in which the internal nodes are boolean expressions and the leaves are the outcomes. The execution of the decision tree traces a path from the root to a leaf. At each internal node, its expression is evaluated, and the left branch is taken if the expression is true; otherwise the right branch is taken.

  i. Describe the decision tree model of comparison sorts.
  ii. What are the expressions in the internal nodes?
  iii. What are the outcomes at the leaves?
  iv. Explain why there are at least $n!$ leaves.

(c) Show an example of the decision tree model for $n = 3$.

(d) The height of the decision tree model represents the worst-case number of comparisons. Show that the height of the decision tree model must be at least $\lg(n!)$.

(e) Show that $\lg(n!)$ is $\Theta(n \lg n)$. Hint: Use the fact that $\lg(mn) = \lg(m) + \lg(n)$.

P2 This problem is concerned with the SELECT algorithm.

1. Write pseudocode for PARTITION(A, p, r) that randomly selects a pivot from the subarray $A[p..r]$, and then rearranges $A[p..r]$ so that $A[q]$ is the pivot, $A[p..q-1]$ contains values less than or equal to the pivot, and $A[q+1..r]$ contains values greater than or equal to the pivot. PARTITION(A, p, r) returns the index q of the pivot.
2. Write pseudocode for SELECT(A, p, r, i) that returns the ith largest number in A using your PARTITION algorithm.
3. Justify and solve a recurrence relation that describes SELECT's asymptotic running time in the best-case.
4. Provide a persuasive informal argument that SELECT's average-case behavior will have the same asymptotic running time.
5. Suppose that we are interested in finding k order statistics, that is, the i_1th largest number, the i_2th largest number, ..., and the i_kth largest number. Revise your SELECT algorithm (call it SELECT-k) so that its average-case running time will be $\theta(n \log k)$ for an array with n elements and the task is to find k order statistics.
6. Provide a persuasive informal argument that your SELECT-k algorithm has $\theta(n \log k)$ average-case running time.