

CS 5150/6150: Assignment 2

Due: Sep 12, 2011

Bethany Azuma

This assignment has ?? questions, for a total of ?? points and ?? bonus points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question 1: Recurrences [??]

Solve each of the following recurrences. You may use any method you like, but please show your work. In each recurrence, you may assume convenient starting values for $T(0)$ or $T(1)$ unless otherwise specified. Note that c is an undetermined constant.

Solving a recurrence means that you provide a bound of the form $T(n) = O(f(n))$ for a specific f . Tight bounds get full credit: for example, if the recurrence is $T(n) = 2T(n/2) + cn$, the answer $T(n) = O(n^2)$ is correct but not precise enough, and you will not get full credit for it.

(a) [4] $T(n) = 3T(n/2) + cn$

Solution: Here each node has 3 children so on the i th level there are 3^i nodes each of size $c\frac{n}{2^i}$. Hence at each level the sum is $c\left(\frac{3}{2}\right)^i$. This gives the level-by-level sum

$$T(n) = \sum_{i=0}^L \left(\frac{3}{2}\right)^i cn$$

This geometric series is dominated by its last term $\left(\frac{3}{2}\right)^i cn$. Assuming each leaf contributes 1, the final term is equal to the number of leaves. The tree has $\log_2 n$ levels the tree has $3^{\log_2 n} = n^{\log_2 3}$ leaves. Hence

$$T(n) = \Theta\left(n^{\log_2 3}\right).$$

(b) [4] $T(n) = 4T(n/4) + n \log n$

Solution: Here the i th level has 4^i nodes each with a value of

$$\frac{n}{4^i} \lg \frac{n}{4^i} = \frac{n}{4^i} (\lg n - \lg 4^i) = \frac{n}{4^i} (\lg n - 2i)$$

The tree has $L = \log_4 n = \frac{\lg n}{2}$ levels which gives a level by level sum of

$$T(n) = \sum_{i=0}^L n(\lg n - 2i) = \sum_{i=0}^{\frac{\lg n}{2}} n(\lg n - 2i)$$

It follows that,

$$\begin{aligned} T(n) &= \frac{\lg n}{2} n \lg n - 2 \sum_{i=0}^{\frac{\lg n}{2}} i \\ &= \frac{\lg n}{2} n \lg n - \frac{\lg n}{2} \left(\frac{\lg n}{2} + 1 \right) \\ &= O(n(\lg n)^2) \end{aligned}$$

(c) [4] $T(n) = 5T(n/8) + \sqrt{n}$

Solution: Here the recurrence fits the Akra-Bazzi form of the master theorem. In particular, $a = 5$, $b = 8$ and $c = \frac{1}{2}$ because

$$\frac{1}{2} < \log_8 5 \approx 0.773976$$

$$T(n) = \Theta(n^{\log_8 5}) \text{ so}$$

$$T(n) = \Theta(n^{\log_8 5})$$

(d) [4] $T(n) = 2T(n/2) + \frac{n}{\log n}$

Solution: Here the sum of the nodes on the i th level is

$$\frac{n}{\lg n - i}$$

and the depth is at most $\lg n - 1$. So,

$$T(n) = \sum_{i=0}^{\lg n - 1} \frac{n}{\lg n - i} = \sum_{j=1}^{\lg n} \frac{n}{j} = nH_{\lg n}$$

where $H_n = \sum_{i=1}^n \frac{1}{i}$. Because $H_n = \Theta(\lg n)$,

$$T(n) = \Theta(n \lg \lg n)$$

(e) [4] $T(n) = \frac{T(n-1)}{T(n-2)}$, $T(0) = 1, T(1) = 2$.

Solution:

$$1 \leq T(n) \leq 2$$

Question 2: Finding a local minimum.....[??]

Suppose we are given an array $A[1..n]$ with the special property that $A[1] \geq A[2]$ and $A[n-1] \leq A[n]$. We say that an element $A[x]$ is a *local minimum* if it is less than or equal to both its neighbors, or more formally, if $A[x-1] \geq A[x]$ and $A[x] \leq A[x+1]$. For example, there are six local minimums in the following array

9	7	7	2	1	3	7	5	4	7	3	3	4	8	6	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

We can obviously find a local minimum in $O(n)$ time by scanning through the array. Describe and analyze an algorithm in $O(\log n)$ time. [Hint: with the given boundary conditions, the array **must** have at least one local minimum. Why?]

Question 3: Searching for a target sum.....[??]

Given an array S of n numbers and a query x , design an algorithm that determines whether there exist two elements of S that sum to x .

Question 4: Hidden Surface Removal.....[??]

You are given n *nonvertical* lines in the plane, with the i^{th} line specified by the equation $y = a_i x + b_i$. Assume that no three lines meet at a single point. A line ℓ_i is *uppermost* at some x -coordinate x_0 if $a_i x_0 + b_i > a_j x_0 + b_j$ for all $j \neq i$. Intuitively, a line is uppermost at x_0 if you can “see” it from above (looking down from (x_0, ∞)).

A line is *visible* if there is some x_0 for which it is uppermost. Output the set of visible lines as efficiently as possible.

Question 5: Selection.....[??]

- (a) [10] Suppose we are given two sorted arrays $A[1..n]$ and $B[1..n]$ and an integer k . Describe and algorithm to find the k th smallest element in the union of A and B in $\Theta(\log n)$ time. For example, if $k = 1$, your algorithm should return the smallest element of $A \cup B$; if $k = n$, your algorithm should return the median $A \cup B$. You can assume that the arrays contain no duplicate elements. [Hint: First solve the special case $k = n$.]
- (b) [5] Now suppose we are given *three* sorted arrays $A[1..n]$, $B[1..n]$, and $C[1..n]$ and an integer k . Describe an algorithm to find the k th smallest element in $A \cup B \cup C$ in $O(\log n)$ time.
- (c) [10] Finally, suppose we are given a two dimensional array $A[1..m][1..n]$ in which every row $A[i][\cdot]$ is sorted, and an integer k . Describe an algorithm to find the k th smallest element in A as quickly as possible. How does the running time of your algorithm depend m ? [Hint: use the linear time SELECT algorithm as a subroutine.]

Question 6: Implementing a high quality selection routine.....[??]

There is often a gap between the theory and practice of algorithm design. We’re going to explore that gap with the help of the selection problem. Our goal will be, given input set of numbers S and a parameter k , to compute the k^{th} largest number in S . Note that the median corresponds to $k = n/2$.

- (a) Implement an algorithm for k -selection.
- (b) Experiment with different choices for finding a pivot. Specifically, use different groupings of the input and report on how this affects the performance
- (c) How does this algorithm compare to sorting and then taking the k^{th} element in practice ? Provide concrete empirical evidence in support of your answer.
- (d) How do random pivots compare to computing deterministic pivots ? Again, experiment with different random pivots and provide concrete empirical evidence in support of your answer.