Homework #3
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2015

Due on: Tuesday, Feb 17th, 5pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On blackboard, under student assessment
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

February 19, 2015

1 Problem 1 (20 points)

Let S be a sorted array of distinct integers $S[1], S[2], \ldots, S[n]$. Design an algorithm to determine whether there exists an index i such at $S[i] = i$. For example, in $1, 2$, $a_2 = 2$. In $2, 1$, there is no such i. You algorithm should work in $O(\log n)$ time. Prove the correctness of your algorithm.

Answer:

Probably, when you get this problem, you can easily come up with an algorithm that is similar to Binary Search. W.L.O.G, we can assume that n is even and the array is in ascending order. Since the array S is sorted and only contains distinct integers, you can first check the index $\frac{n}{2}$ with the following three cases: if the value of $S[\frac{n}{2}]$ is exactly $n/2$, return TRUE; if $S[\frac{n}{2}] > \frac{n}{2}$, then try to solve the subproblem on the left half side of the array; if $S[\frac{n}{2}] < \frac{n}{2}$, then try to solve the subproblem on the right half side of the array. Recursively execute like a binary search until there is no element in the array can be checked. If no TRUE returned before, return FALSE.

Time Analysis:

Since $T(n) = T(n/2) + O(1)$, by the case 2 of the master theorem, $T(n) =$

$O(\log n)$.

Correctness:

Please refer to the proof of correctness for Binary Search or similar. Induction proof is an easier way to show the correctness of the proof. Other proofs are welcome.

2 Problem 2 (20 points)

Resolve the following recurrences. Use Master theorem, if applicable. In all examples assume that $T(1) = 1$. To simplify your analysis, you can assume that $n = a^k$ for some a, k.

1. $T(n) = 2T(n/2) + n^4$
   Answer: $a = 2, b = 2, \log_b^a = 1 < 4$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n^4)$.

2. $T(n) = T(n/6) + n^{0.001}$
   Answer: $a = 1, b = 6, \log_b^a = 0 < 0.001$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n^{0.001})$.

3. $T(n) = T(\sqrt{n}) + 4$
   Answer: Assume $n = a^k$, $T(a^k) = T(a^{\frac{k}{2}}) + 4$. Let's denote $T(a^k) = A(k)$ and replace $T(n)$ with $A(k)$. Thus, $A(k) = A(k/2) + 4$. Now we can use Master Theorem, $a = 1, b = 2, \log_b^a = 0$. By applying the case 2 of Master Theorem, $A(k) = \Theta(\log k)$. Since $k = \log_a^n$, $T(n) = \Theta(\log \log_a^n)$.

4. $T(n) = 6T(n/5) + n$
   Answer: $a = 6, b = 5, \log_b^a > 1$. By applying the case 1 of Master Theorem, $T(n) = \Theta(n^{\log_5^6})$

5. $T(n) = 6T(n/200) + n^{200}$
   Answer: $a = 6, b = 200, \log_b^a < 200$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n^{200})$

6. $T(n) = n + T(n/3)$
   Answer: $a = 1, b = 3, \log_b^a = 0 < 1$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n)$

7. $T(n) = 2T(n/2) + n^2$
   Answer: $a = 2, b = 2, \log_b^a = 1 < 2$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n^2)$

8. $T(n) = nT(n/6)$

   Answer: Assume $n = 6^k$, $T(n) = 6^k \cdot 6^{k-1} \cdot 6^{k-2} \cdot \dots \cdot T(6^k/6^k) = 6^{\frac{k^2+k}{2}}$. So $T(n) = n^{\frac{\log_6^n + 1}{2}}$.

9. $T(n) = 8T(n/4) + n$
   Answer: $a = 8, b = 4, \log_b^a > 1$. By applying the case 1 of Master Theorem, $T(n) = \Theta(n^{\log_4^8})$

10. $T(n) = T(n/3) + n^2$
    Answer: $a = 1, b = 3, \log_b^a = 0 < 2$. By applying the case 3 of Master Theorem, $T(n) = \Theta(n^2)$

3 Optional Exercises

Solve the following problems and exercises from CLRS: 4-3, 4-1, 7-3.