Algorithms Middle Examination

Nov. 15, 2013 (10:10 ~ 12:20)

1. (8%) (Chapter 4) Give asymptotic upper and lower bounds for $T(n) = T(\sqrt{n}) + 1$. Assume that $T(n)$ is constant for $n \leqq 2$.

2. (8%) (Divide-and-Conquer) Suppose we have 15 identical looking coins numbered 1 through 15 and only one of them is counterfeit (偽造) coin whose weight is different with the others. Suppose further that you have one balance scale (天平秤). Develop a method for finding the counterfeit coin with minimum number of weighings (量稱).

3. (6%) (Chapter 6) What is the running time of Heapsort on an array of length n that is already sorted in increasing order? What about decreasing order? (You need use a few sentences to explain your answer.)

4. (10%) (Chapter 7) Suppose that the splits at every level of quicksort are in the proportion $1-\alpha$ to α, where $0 < \alpha \leq 1/2$ is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately $-\lg n/\lg\alpha$ and the maximum depth is approximately $-\lg n/\lg(1-\alpha)$. (Don't worry about integer round-off.)

5. (10%) (Chapter 7) What is the running time of quick sort algorithm when all elements of array A have the same value? How to improve the performance of quick sort algorithm when there are many elements have the same value?

6. (5%) (Chapter 8) Please give a decision tree for insertion sort operating on three elements a, b, and c.

7. (5%) (Chapter 8) It is known that $\Omega(n\log n)$ is a lower bound for sorting. However, we have seen algorithms like counting sort or radix sort which can sort n items in $O(n)$ time. Is there a contradiction? If not, why? Explain?

8. (10%) (Chapter 9) Show that the second largest of n elements can be found with $n + \lceil \lg n \rceil - 2$ comparisons in the worst case.

9. (10%) (Chapter 9) Let $T(n) = O(n) + 1/n \sum_{1 \le k \le n} T(\max\{k\text{-}1, n\text{-}k\})$. Please prove $T(n) = O(n)$. (Hint: Use mathematic induction method.)

10. (10%) (Chapter 15) What are the two key ingredients that an optimization problem must have in order for dynamic programming to apply? Please give an example that the dynamic programming cannot be applied.

11. (10%) (Chapter 15) Determine the cost and structure of an optimal binary search tree for a set of n = 5 keys with the following probabilities:

i	0	1	2	3	4	5
p_i		0.15	0.05	0.2	0.05	0.1
q_i	0.1	0.1	0.05	0.05	0.1	0.05

12. (10%) There is a staircase with n steps. Your friend, Jack, wants to count how many different ways he can walk up this staircase. Because John is quite tall, in one move, he can choose either to walk up 1 step, 2 steps, or 3 steps. Let F_k denote the number of ways John can walk up a staircase with k steps. Derive a recurrence for F_k, and show that F_n can be computed in O(n) time.