Algorithms First Examination

(2021/10/27)

(**Note that, if you design an algorithm, you must have pseudo code to show your algorithm and analyze the algorithm time complexity. It would help if you put comments after your pseudocode to clarify your algorithm.**)

1. (Chapter 2 Exercise) (10%)(**You need to explain your answer briefly. Otherwise, you cannot get a score.**)
   (a) (5%) Suppose we have 31 identical-looking balls numbered 1 through 31, and only one of them is a counterfeit ball whose weight is different from the others. Suppose further that you have one balance scale. What is the minimum number of weighing times required to find the counterfeit ball in the worst case?
   (b) (5%) Someone designs his/her sorting algorithm as follows. Can the code work in any situation? If it can, briefly explain its correctness. If it cannot, give a counterexample.

```
Sorting (A)
for i = 1 to A.length - 1
    for j = A.length downto i+1
        if A[j] < A[j-1]
            exchange A[j] with A[j-1]
```

2. (Chapter 3 Exercise) (10%) Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove or disprove that

   (a) $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

   (b) $f(n) = \Theta(f(n/2))$.

3. (Chapter 4 Exercise) (10%) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n-1) + T(n/2) + n$. Use the substitution method to verify your answer.

4. (Chapter 4 Exercise) (10%) How would you modify Strassen's algorithm to multiply n x n matrices in which n is not an exact power of 2? Show that the resulting algorithm runs in time $\theta(n^{\lg 7})$.

5. (Chapter 6 Exercise) (10%) Please give an example to show how to implement a first-in, first-out queue with a priority queue. Please provide an example of implementing a stack (first-in, last-out) with a priority queue.

6. (Chapter 7 Homework) (10%) Use the substitution method to prove that the best-case time complexity of QUICKSORT is Ω (n lg n).

7. (Chapter 7 Slides) (10%) Please implement an in-place partition algorithm for Quicksort. For an array A[p .. r], the in-place partition algorithm returns an index q such that each element of A[p .. q-1] is less than or equal to A[q], and each element of A[q+1.. r] is greater than A[q].

8. (Chapter 8 Homework) (10%) Given a non-negative integer array A of length N. Please design an $O(N)$ algorithm and write the pseudocode to find the largest X that satisfies the equation: at least ($\geq$) X integers in A are larger than X. Briefly explain your pseudocode and show it is $O(N)$.

9. (Chapter 8 Slides) (10%) Assume an array A[1..n] of n elements, and each element is drawn uniformly at random from the interval [0,1). If we use the bucket sort to sort array A, please prove that bucket sort's average case running time is $\theta(n)$.

10. (Chapter 9 2020 Exam.) (10%) SELECT works with groups of 5 elements. Prove that SELECT cannot work in worst-case linear time with groups of 3 elements with the following questions.
    a. (3%) After choosing the median of medians, divide the original group into X and Y. $\text{Max}(X, Y) = ?$
    b. (3%) $T(n)$ =? Briefly explain each term.
    c. (4%) Disprove by assuming $T(n) \leq cn$ and getting a conflict.