南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

89

Student ID: 12312710 Student Name: 肖炜翔

CS203 Data Structure and Algorithm Analysis Quiz 1

**Note 1:** Write all your solutions in the question paper directly. You can ask additional answer paper if necessary

**Note 2:** If a question asks you to design an algorithm, full marks will be given if your algorithm runs with optimal time and space complexity

**Note 3:** If a question asks you to design an algorithm, you should **first** describe your ideas in general words, **then** write the pseudocode, and **end** with time complexity analysis.

**Problem 1 [20 points]** Recursive Algorithm.

**Factorial** (int n)

1. if n ≤ 1 then
2. return 1
3. *else*
4. *return n** **Factorial** (n-1)

5040 ÷2 → 2520 ÷3 → 840 → 6×4×5=120.
120×6=720
720×7=5040.

20

(a)[4 points] The time complexity of the algorithm is O(n)

(b)[4 points] Let n = 3, write down its output: 6

(c)[4 points] Let n = 5, write down its output: 120

(d)[8 points] Please modify the above pseudocode such that the output is "5040" when n=6. (You only can use the used functions and statements)

```
Factorial (int n)
    if n≤0 then
        return 1
    else
        return (n+1) * Factorial (n-1)
```

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

## Problem 2 [30 points] Element Shifting Problem

Given an array A with $n$ integers and an integer t ($t < n$). Design an algorithm to shift the sequence in A by t positions. For example, if A = {10, 7, 12, 18, 16, 20, 30} and t=3, then the output of your algorithm should be {18, 16, 20, 30, 10, 7, 12}.

Please note you only can O(1) extra-memory for this problem.

由于我不能创造其它额外的数组来存，因此我只能对数组进行平移
注意到此数组平移了 t 次，则 "shift" 操作要进行 t 次。
对数组平移

```
Shift ( int[] A, int t)
    for i←1  to t
        int temp ← A[ A.length]
        for  j← A.length -1.  j decrease to 1
            A[j+1] ← A[j]
        A[1] ← temp
    return A
```

→ 执行对数组平移的操作了 $t$ 次，

而平移一次操作了 n 次

∴ 时间复杂度为 O(tn)

27

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

## Problem 3 [20 points] Find the k-th largest number

Given an unsorted array A with n integers and an integer k, we guarantee all integers are distinct and 1<= k <= n, please design an algorithm to find the k-th largest number in it.

Example: suppose A = [4,3,5,1,2], when k=1, the 1-st largest number is 5, when k=4, the 4-th largest number is 2.

14

(1) 先遍历加(k-1)次数组，使每一次遍历后中，最大的数被赋值成为 Integer.minimum(-2147483648)，则在第k次遍历时，最大的数就是 k-th largest number.

```
kthLargestNum( int[] A, int k)
    for i←1 to (k-1)
        max ← Integer.minimum
        index ← 1
        for j←1 to A.length
            if A[j] ≥ max
                max ← A[j]
                index ← j
        A[index] ← Integer.minimum.
    kMax ← Integer.minimum
    for j←1 to A.length
        if A[j] ≥ kMax
            kMax ← A[j]
    return kMax
```

方法二：排序后输出.

(2) kMethodTwo(int[

MergeSort(A)

return A[n-

分析：MergeSort 用了 nlogn

∴O(nlogn)

当k很大时，方法二更加快

分析：

在(k-1)次遍历中，

最后再用了 A.length 次对比.

∴ 共用了 k·n 次对比

∴ O(kn)

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

**Problem 4 [30 points] Filling blank questions**

(a) [5 points] Please proof $2n^2 + 5n\log n + 10 = O(n^2)$ by using the Big-O definition. to find a,b such that: $2n^2+5n\log n+10 \le a\cdot n^2$ when $n>b$

we find that when $a=10000, b=10000$, it satisfied.

(b) [5 points] Suppose the sequence 11, 12, 13, 7, 8, 23, 4, 5 is the immediate sorting result after the second iteration of a sorting algorithm. Which sorting algorithm is used: B.

A. Bubble sort B. Insertion sort C. Selection sort D. Quick sort

(c) [5 points] Which of the following function is not $O(n^3)$ ( D )

A. $67n^2\log n$ B. $(\log_2 n)^{98}$ C. $938593729n^{2.9}$ D. $n^{3.1}/\log^2 n$

(d) [8 points] considering the combine function in the merge sort algorithm, it combines two sorted sub-array (i.e., L and R) into one sorted array A. Suppose the length of L and R is m and n, respectively.

(1) How many times of comparison will be used? $[\min(m,n) \sim m+n]$

(2) What is the time complexity of the combine function? $O(m+n)$

(e) [7 points] Given an unsorted array A with n integers, please describe your idea (and its time complexity) to find the pair of A[i] and A[j] such that |A[i] – A[j]| is minimal. (You do not need write pseudocode for it)

① 用 merge sort 将数组升序排序

② 遍历一遍数组，并记录 A[n] - A[n-1]

A[n-1] - A[n-2]

...

A[2] - A[1] 的最小值，此值为答案

分析：由①排序用 $3n\log n$ 的时间

②遍历用 $n$ 的时间 $\therefore n+n\log n = n(\log n+1) = O(n\log n)$