

COMP 3761 Assignment 1

Due: Wednesday **Jan 21, 2015** at 6:30pm

1. Consider the following algorithms:

```
(A) long partA(int n)
    {
        long sum = 0;
        for (int i = 0; i < n; i++)
            sum = sum + (i * i + 1) * (i * i + 1);
        return sum;
    }
```

```
(B) long partB(int n)
    {
        long sum = 0;
        for (int i = 1; i <= n; i++) {
            int m = 1;
            for (int j = 1; j <= i - 1; j++) {
                m = m * 2;
            }
            sum = sum + (i + 1) * m;
        }
        return sum;
    }
```

```
(C) long partC(int n)
    {
        long sum = 0;
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                for (int k = 1; k <= n; k++)
                    sum = sum + i * j * k;
            }
        }
        return sum;
    }
```

```
(D) long partD(int n)
    {
        long sum = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < i; j++)
                sum = sum + (i + j + 1);
        }
        return sum;
    }
```

Assume that the input value n is a positive integer.

- (1.1) Use summation notations to show what each algorithm computes for any input integer $n \geq 0$.
[1 point x 4]
- (1.2) State which arithmetic operation is considered as the basic operation for analyzing the running time efficiency class in each algorithm. [0.5 point x 4]
- (1.3) Count how many times the basic operation is executed for the input size n in each algorithm. Express the count as an exact function of n . Show your steps to support your answer.
Hint: You need to apply the proper summation formulas from Appendix A in the textbook.
[2 points x 4]
- (1.4) What is the time efficiency class of each algorithm? [0.5 point x 4]

2. Problem: Inversion Count

Let $A[0..n-1]$ be an array of n integer numbers. Let i and j be any arbitrary indices in the array A , where $0 \leq i, j \leq n-1$. A pair of array values $(A[i], A[j])$ is said to be an **inversion** if the values $A[i]$ and $A[j]$ are out of order with respect to their index positions i and j . That is, if $i < j$ but $A[i] > A[j]$, then the pair $(A[i], A[j])$ is an inversion.

For example, given an array $\{1, 3, 2\}$, the pair of numbers $(3, 2)$ is considered as an inversion.

Sample input file: IntegerArray_10000.txt.

The test file contains 10000 distinct integers between 1 and 100,000 (inclusive) in arbitrary order. The entire file represents one input integer array. Each line contains one single integer number, where the i th row of the file indicates the i th entry of an array.

- (a) Design an algorithm to count the number of inversions in a given array with n numbers. Provide the pseudocode of your solution. [2]
- (b) Analyze the worst-case time efficiency class of your algorithm. Show your mathematical steps to support your answer. [2]
- (c) Implement your algorithm of InversionCount in Java. Count both the number of inversions and the actual number of element comparisons performed when the algorithm is executed. [5]
- (d) Your test program must read the input file name from the standard input (keyboard). Write a method that prompts the user for the test input file name and reads in all the integers from the user-provided input file and constructs an input array instance for the problem of Inversion Count. You may assume that all input files have the correct file format as specified above. [5]
- (e) Test your solution with the sample input. Print both the number of inversions and the actual number of element comparisons executed for the given input. [2]

Important Note: Before you start working on this assignment, please read carefully the document (Comp3761_AssignmentGuidelines.pdf) about the general guidelines for the course assignments.

Ensure to follow the detailed instructions on the assignment submission.