Problem 0. Information (worth 1 point)

Write your name and KAIST ID at the top of **EACH** page on your answer paper.

Problem 1. (5 points @ each)

For **EACH** of the functions below,

1. Give the order of growth in terms of n, using the big-Oh notation. (worth 2 points)
2. Explain why it takes that running time. (worth 3 points)

1)

```
public static int p1(int n) {
     int i = 0;
     for (int j = 0; j < n; j++)
         i++;
     return i;
 }
```

2)

```
public static int p2(int n) {
        int i = 0;
     for (int j = 0; j < n; j++)
         for (int k = 0; k < j*j; k++)
             i++;
         return i;
 }
```

3)

```
public static void p3 (int n) {
     if (n == 0) return;
        p3(n-1);
        p3(n-1);
  }
```

Problem 2. (7 points @ each)

1) Using the 64-bit memory cost model from lecture, how many bytes does **an object of the Memory class** in the following code use? Provide ALL intermediate steps that show how you compute the memory usage.

```
public class Memory{
         private double a;
         private String b;
         private int[] data;

}
```

2) Using the 64-bit memory cost model from lecture, how many bytes does **an object of the Node class** in the following code use? Provide ALL intermediate steps that show how you compute the memory usage.
   - Notes: When an object is referenced, do not count the memory for that object, but do include the memory for the reference to the object. The Node class is a top-level class, not a nested class in this question.

```
public class Node{
        private int n;
        private int m;
        private List list;
        private Node prev;
}
```

Problem 3. (10 points)

Suppose that you implement Stack ADT using a dynamic array with a doubling-and-halving strategy that we studied in class.

1) Show ALL intermediate steps that present how we push 1, 2, 3, 4, 5, 6, 7, 8 in that order into the Stack described above, drawing a resulting array after each push operation with a concise explanation. Suppose that we push items into an initially empty array of size one. (worth 8 points)
2) How many times is an array resized if we perform N push operations into the Stack described above? (worth 2 points)

Problem 4. (10 points)

Given an array of integers, you are required to design an efficient algorithm that finds the largest value of multiplying two integers in the array. Array elements are in random order in the array. Integers in the array are distinct from one another.

For example, if we have an array {-5, 3, -1, 0, -4}, two integers -5 and -4 produce the largest value of multiplication, which is 20.

- Notes: Your algorithm will be graded on correctness, efficiency, and clarity.

1) Explain an efficient algorithm that performs the task described above. (worth 5 points)
2) What is the running time of your efficient algorithm if the array has n elements? (Use big-Oh notation)—worth 3 points. Explain why it takes that running time.—worth 2 points