Design and Analysis of Computer Algorithms

Homework #1: Due 11:59PM, 13th March 2022 (Sunday).

Problem #1 (15 points). Programming

Shopping Addiction

Kyunghee Jung is addicted to shopping. Whenever a shop has discount events, she completely goes crazy and wants to buy all items. You are her boyfriend; you cannot stop her from shopping, but you can suggest her a good shopping strategy to save her money. On this valentine's day, a shop offers a very good deal "**Buy 2, get 1 free**" with a rule that in one bill, **only the cheapest ones get free**. Your task is to help her find the maximum discount she can get.

Example:

- Your girlfriend wants to buy 7 items, costing $35, $40, $30, $10, $15, $20, and $25.
- If she buys all items in one bill, she gets 2 free items which are the cheapest ones of $10 and $15. Consequently, she gets the discount of $25 and must pay $150.
- If she buys those 7 items by 3 separated bills, she may get a bigger discount. For instance:
    - The first bill: 3 items $40, $30, and $25 → $25 discount.
    - The second bill: 3 items $35, $20, and $10 → $10 discount.
    - The third bill: 1 item $15 → no discount.
    - Eventually, she earns a total discount of $35 and must pay only $140.

Input is read from the text file **input.txt** consisting of:

- The first line is the number of items she buys
- The second line is the list of prices of the items

Output is written to the text file **output.txt** consisting of **ONLY ONE NUMBER** which is the maximum discount she can get.

Example:

input.txt	output.txt
6 10 40 20 35 30 25	40

Limitation:

- 1 ≤ number of items she buys ≤ 200,000
- 1 ≤ the cost of an item ≤ 1,000,000
- Processing time of the proposed algorithm ≤ 1 second

Notice:

- Using standard C, C++, Java, Python (equivalent)
- The name of the source code file is **shopping.xxx** (The extension **xxx** depends on the used programming language)
- The name and format of input and output files **must follow exactly what was described in the problem**. Students **get zero point if they do not follow the format**
- The **output file only has one value**
- The main function must follow the template in case you use C/C++:

```
int main(int argc, const char* argv[])
{
        //content
        …
        return 0;
}
```

- Sample code for writing to a text file (C/C++):

```
#include <stdio.h>
int main()
{
    int num;
    FILE *fptr;
    fptr = fopen("output.txt","w");
    printf("Enter num: ");
    scanf("%d",&num);
    fprintf(fptr,"%d",num);
    fclose(fptr);
    return 0;
}
```

- Sample code for reading a text file (C/C++):

```
#include <stdio.h>
int main()
{
    int num;
    FILE *fptr;
    fptr = fopen("input.txt","r");
    fscanf(fptr,"%d", &num);
    printf("Value of n=%d", num);
    fclose(fptr);
    return 0;
}
```

Hint:

There are 2 cases:

- If the number of items he buys < 3, the total discount is 0
- Otherwise, sorting the costs of items in descending order, then splitting them into groups of 3 and adding up values of the third item of each group.

Problem #2 (15 points). Programming

Triangle Counting

Given N points with corresponding x and y coordinates on the Cartesian coordinate system.

Task:

Checking how many isosceles or equilateral triangles can be formed from the given N points?

Input is read from the text file **input.txt** consisting of:

- The first line is the number of points N
- The next N lines are the coordinates of N points (each line contains the x and y coordinates of one point)

Output is written to the text file **output.txt** consisting of ONLY ONE NUMBER which is the maximum number of isosceles or equilateral triangles can be formed.

Example:

input.txt	output.txt
3 0 3 1 0 2 3	1

Limitation:

- $N \leq 100$
- $|x|, |y| \leq 10^9$
- Processing time of the proposed algorithm ≤ 1 second

Notice:

- Using standard C, C++, Java, Python (equivalent)
- The name of the source code file is **triangles.xxx** (The extension **xxx** depends on the used programming language)
- The name and format of input and output files **must follow exactly what was described in the problem**. Students **get zero point if they do not follow the format**

Hint:

- Using 3 loops to check all possible sets of 3 points then check if each set of 3 points can form an isosceles or equilateral triangle.
- Be careful with the case that 3 points have same coordinates or are on a line.

Problem #3 (15 points). Suppose we are comparing two sorting algorithms.

a) Suppose that for all inputs of size n, the first algorithm runs in $8n^2$ seconds, while the second algorithm runs in $64n\log_2 n$ seconds. For which values of n does the first algorithm beat the second algorithm?

b) What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n?

Problem #4 (15 points). We are sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with $A[1]$. Then, find the second smallest element of A, and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A.

a) Write pseudocode for this algorithm, which is known as SELECTION sort.

b) Why does it need to run for only the first $n-1$ elements, rather than for all n elements?

c) Give the best-case and worst-case running times of selection sort in Θ-notation.

Problem #5 (10 points). Prove by induction on $n \geq 1$ that $\sum_{i=1}^{n} 1/2^i = 1 - 1/2^n$.

Problem #6 (15 points). Conceptually, a recursive merge sort works as follows:

- Divide the unsorted list into 2 sub-lists, each containing a half of the original list elements.
- Sort the two sub-lists by merge sort
- Merge the two sub-lists into one list

a) Write a recurrence for the running time of this recursive version of merge sort.

b) Solve the recurrence equation

Problem #7 (15 points). For each of the following pairs of functions, either $f(n)$ is in $O(g(n))$, $f(n)$ is in $\Omega(g(n))$, or $f(n) = \Theta(g(n))$. Determine which relationship is correct and briefly explain why.

a) $f(n) = logn^2; g(n) = logn + 5$

b) $f(n) = \sqrt{n}; g(n) = logn^2$

c) $f(n) = log^2 n; g(n) = logn$

d) $f(n) = n; g(n) = log^2 n$

e) $f(n) = nlogn + n; g(n) = logn$

f) $f(n) = 10; g(n) = log10$

g) $f(n) = 2^n; g(n) = 10n^2$

h) $f(n) = 2^n; g(n) = 3^n$

i) $f(n) = n^2 + 3n + 4; g(n) = 6n + 7$

j) $f(n) = n\sqrt{n}; g(n) = n^2 - n$

What you have to submit:

1) Your source programs and executable files.
2) Your input data file and output files (The graders will test your program by his input data file).
3) Documentation file. (**HW1.DOCX**)
   - Solution of the assigned problems.
   - Write the explanation about your implementation.

◆ Submit your compressed file named as HW1_ID_NAME.zip (ex. HW1_2013711123_홍길동.zip) to iCampus.

NOTICE:

- ✓ BOTH ORIGINAL AND COPY WILL GET –30 POINTS EACH INSTEAD OF 0S.
- ✓ ANY SOURCE CODE WITH COMPILE OR RUNTIME ERROR WILL GIVE YOU 0 POINTS.
- ✓ THERE WILL BE POINTS OFF FOR INAPPROPRIATE SUBMISSION STYLE.
- ✓ ALL THE HOMEWORK MATERIALS (INCLUDING EMAIL CONTENTS AND DOCUMENTATION) SHOULD BE MADE IN **ENGLISH**.

Good luck!