

## Homework #1

Due Time: 2018/10/18 (Thu.) 14:20

Contact TAs: [ada-ta@csie.ntu.edu.tw](mailto:ada-ta@csie.ntu.edu.tw)

### Instructions and Announcements

- There are **three programming problems** and **two hand-written problems**.
- **Programming.** The judge system is located at <https://ada18-judge.csie.org>. Please login and submit your code for the programming problems (i.e., those containing “Programming” in the problem title) by the deadline. **NO LATE SUBMISSION IS ALLOWED.**
- **Hand-written.** For other problems (also known as the “hand-written problems”), please turn in a **printed/written version** of your answers to the instructor at the beginning of the class. In case that your homework is lost during the grading, you can also upload your homework to the NTU COOL system. **NO LATE SUBMISSION IS ALLOWED.**
- **Collaboration policy.** Discussions with others are strongly encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (e.g., the Internet URL you consulted with or the people you discussed with) on the first page of your solution to that problem. You may get zero point due to the lack of references.

## Problem 1 - ADA Coin (Programming) (10 points)

### Problem Description

Nowadays cryptocurrency is the hottest topic and Alpha noticed an unpopular one: ADA coin. After the first ADA class, Alpha was surprised by the popularity of ADA, so he believes that a cryptocurrency named “ADA” will become the dominating coin on the earth soon.

Based on the knowledge from “Machine Learning Foundations” and “Machine Learning Techniques”, Alpha has built a *DEEP* learning model to predict the price trend of the ADA coin for next  $n$  days. That is, Alpha believes he knows the price change of the ADA coin for the next  $n$  days.

Alpha wants to earn as many dollars as possible, but due to the high transaction fee, he decides to *buy and sell exactly once*. Also, Alpha does not want to buy and sell at the same day, so it is possible that Alpha loses some money even if he has known the price trend. Can you write a program to help Alpha find out how much he can earn during the next  $n$  days?

### Input

The first line of the input file contains an integer indicating  $n$ .

The next line contains  $n$  integers, separated by spaces. For the  $i$ -th number  $a_i$ ,  $a_i > 0$  indicates that the price of the ADA coin rises  $a_i$  dollars at the  $i$ -th day, while  $a_i < 0$  indicates that the price of the ADA coin falls  $-a_i$  dollars at the  $i$ -th day.  $a_i = 0$  indicates that the price stays the same.

- $1 \leq n \leq 200000$
- $|a_i| \leq 10000$

### Output

Please output an integer indicating the maximal money Alpha can earn in the next  $n$  days.

#### Sample Input 1

```
6
2 -4 6 -1 0 2
```

#### Sample Output 1

```
7
```

#### Sample Input 2

```
3
-4 -6 -1
```

#### Sample Output 2

```
-1
```

## Problem 2 - Rectangle Area (Programming) (20 points)

### Problem Description

For any 2 points on the 2-dimensional plane, we can form a rectangle, where the edges are parallel to x-axis or y-axis and the diagonal line connects 2 points. That is, with 2 points,  $(x_1, y_1)$  and  $(x_2, y_2)$ , we can define a rectangle with width  $|x_1 - x_2|$  and height  $|y_1 - y_2|$ , and thus its area is  $|x_1 - x_2| \times |y_1 - y_2|$ .

Given a set of points  $S$ , there are  $\frac{|S| \times (|S| - 1)}{2}$  formed rectangles. Can you find the sum of the areas of all rectangles?

### Input

The first line of the input file contains an integer indicating  $|S|$ .

For the next  $|S|$  lines, each line contains 2 integers  $x, y$  indicating a point in  $S$ .

- $2 \leq |S| \leq 100000$ .
- For all  $(x, y) \in S$ ,  $0 \leq x, y \leq 10000$ .

### Subtask 1 (20 %)

- $|S| \leq 1000$

### Subtask 2 (20 %)

- $x, y \leq 100$

### Subtask 3 (60 %)

- No other constraints.

### Output

Please output an integer indicating the answer.

Note that the answer might exceed the capacity of a 32-bit integer.

### Sample Input 1

```
3
1 1
2 3
4 0
```

### Sample Output 1

```
11
```

### Sample Input 2

```
4
1 1
1 1
1 3
3 3
```

### Sample Output 2

```
8
```

## Problem 3 - Joe's Game (Programming) (20 points)

### Problem Description

There are  $n$  piles of stones on the ground. The  $i$ -th pile contains  $a_i$  stones. Your friend, Joe, designs a game and wants to play this game with you.

The game consists of several rounds. In each round, Joe will give you a magic number  $k$ , and then you have to answer how many pairs of piles  $(i, j)$  such that  $a_i + a_j = k$  and  $i < j$ .

### Input

The first line contains an integer  $n$  indicating the number of piles, where  $0 \leq n \leq 1,000,000$ .

The second line contains  $n$  integers  $a_i$  indicating the number of stones of the  $i$ -th pile, where  $0 \leq a_i \leq 100,000$ .

The third line contains an integer  $m$  indicating the number of queries, where  $0 \leq m \leq 1,000,000$ .

The fourth line contains  $m$  integers  $k_i$  indicating the magic number of the  $i$ -th query, where  $0 \leq k_i \leq 1,000,000,000$ .

### Output

Print  $m$  integers in one line — for each answer to each round.

Or, you can print "joe59491" to get a wrong answer.

Note that the answer might exceed the capacity of a 32-bit integer.

#### Subtask 1 (20 %)

- $n \leq 1000$

#### Subtask 2 (20 %)

- $a_i \leq 1000$

#### Subtask 3 (60 %)

- No other constraints.

#### Sample Input 1

```
10
5 2 2 4 1 2 9 0 1 3
7
1 2 3 4 5 6 59491
```

#### Sample Output 1

```
2 4 7 6 6 5 0
```

## Problem 4 - Time Complexity & Arvin's Trip (Hand-Written) (25 points)

- (1) Given each pair of expressions A and B in the following table, please find the relation between them (A is  $O$ ,  $o$ ,  $\Omega$ ,  $\omega$ ,  $\Theta$  or None of B). Assume that  $k \geq 1$ ,  $\epsilon > 0$  and  $c > 1$  are constants. Your answer should be in the form like below:

Given  $A = x^2 + 2$  and  $B = x^2$   
Ans: A is  $O$ ,  $o$ ,  $\Omega$ ,  $\omega$ ,  $\Theta$ , None of B

	A	B
(a) (5%)	$\log_2 n^k$	$n^\epsilon$
(b) (5%)	$\sqrt{n}$	$n^{\sin n}$
(c) (5%)	$2^{\sqrt{\log n}}$	$n \log n$
(d) (5%)	$\log n!$	$\log n^n$

- (2) Arvin is an idol in the CSIE department, and some students who respect him form a group called "A127s". Today, Arvin and A127s are now in the Fabulous Fox Theatre, which is lavishly located in the Atlanta with a 4,665-seat auditorium. The A127s sit in a line of seats. However, there is a problem that all boys sit together and all girls do, too. It is important for Arvin to resolve the issue so that he can match boys and girls.

Now, we give you an id sequence, which denotes the sitting positions of all A127s with  $n$  elements ( $n/2$  boys and  $n/2$  girls). That is, given the static form  $[b_1, b_2, b_3, \dots, b_{n/2}, g_1, g_2, g_3, \dots, g_{n/2}]$ , you need to switch the elements such that the sequence becomes  $[b_1, g_1, b_2, g_2, \dots, b_{n/2}, g_{n/2}]$  with the extra space  $O(\log n)$  (note that  $n$  is even).

- (a) (25%) Give the pseudo code for a Divide-and-Conquer algorithm with  $O(n \log n)$  time complexity that changes the original sequence to the target sequence, and briefly explain your code. Assume that the original ids are stored in the array  $M[0 \dots n - 1]$ .
- (b) (15%) Please show the time complexity of your algorithm using  $\Theta$  and briefly explain how you derive this bound.
- (3) It is time for them to take a picture with this magnificent structure. A127s still stand in a horizontal line, and Arvin needs some of A127s form a second line behind. However, due to Arvin's insistence, the second line needs to be first increasing and then decreasing. It is called "unimodal." Arvin wants to know how many combinations of the A127 that can fit Arvin's requests.

Given a height array  $H[0 \dots n - 1]$  and the number  $k$ . Find the number of the unimodal subsequences with size  $k$  in the array. For instance,

Input:  $H = [142, 188, 189, 163]$ ,  $k = 3$

Output: 3 ( $[142, 188, 163]$ ,  $[142, 189, 163]$ ,  $[188, 189, 163]$  are the unimodal subsequences with size 3.)

- (a) (25%) Give the pseudo code for a Dynamic Programming algorithm with time complexity  $O(kn^2)$  and briefly explain your code. Assume that the original heights are stored in the array  $H$ .
- (b) (15%) Please show the complexity of your algorithm using  $\Theta$  and briefly explain how you derive this bound.

## Problem 5 - Recurrence & ADA-TA-QQ Problem (Hand-Written) (25 points)

- (1) (50%) Solve the following recurrences. Assume that  $T(n) = 1 \forall n \leq 2$ . Write down your proofs thoroughly. If you apply some theorems that are not taught in class, you have to prove them first.

- (a) (10%)  $T(n) = 0.5T(\frac{n}{2}) + n^3$   
Prove or disprove that  $T(n) = \Theta(n^3)$ .
- (b) (10%)  $T(n) = 7T(\frac{n}{4}) + 16n^2 \log n$   
Prove or disprove that  $T(n) = \Theta(n^2 \log n)$ .
- (c) (10%)  $T(n) = 4T(\frac{n}{2}) + n^2 \log n$   
Prove or disprove that  $T(n) = \Theta(n^2 \log^2 n)$ .
- (d) (10%)  $T(n) = T(\frac{n}{2}) + T(\frac{n}{3}) + T(\frac{n}{6}) + n$   
Prove or disprove that  $T(n) = \Theta(n \log n)$ .
- (e) (10%)  $T(n) = \sqrt{n}T(\sqrt{n}) + n$   
Prove or disprove that  $T(n) = \Theta(n \log \log n)$ .

- (2) (50%) Imagine that you are a TA of ADA that is responsible for handling additional enrollment. However, obviously, the number of students who want to register is far more than the classroom capacity.

After giving most of the enrollment sheets to those with higher priorities, you notice that there is only 1 seat left whereas  $n$  students are still waiting. The professor plan to choose the last one based on a raffle: she picks a lucky number  $k$  ( $1 \leq k \leq n$ ) and tells you to choose the student with  $k^{\text{th}}$  largest student ID. We call it “ADA-TA-QQ Problem”.

All people in the class are watching you, making you very nervous. As a TA, you have to solve this very quickly. Unfortunately, the waiting students are not sorted in IDs. Furthermore, you cannot memorize all of their IDs and thus derive the answer easily. What you can do is just comparing IDs or dividing them into groups.

- (a) (15%) Please design a Divide-and-Conquer algorithm that solves ADA-TA-QQ Problem in  $O(n)$  time and  $O(n)$  space. Give the pseudo code and explain your code clearly.
- (b) (10%) Prove the correctness of your algorithm.
- (c) (15%) Analyze your algorithm, showing that it runs in  $O(n)$  time.
- (d) (5%) Analyze your algorithm, showing that it runs in  $O(n)$  space.
- (e) (5%) Justify that the time complexity of ADA-TA-QQ Problem is  $\Theta(n)$ .