Homework #3

Due Time: 2015/11/9 (Mon.) 12:00
Contact TAs: ada@csie.ntu.edu.tw

Instructions and Announcements

- For the first time submitting your ADA 2015 HW, please create a repository for ADA on bitbucket (https://bitbucket.org) and share the repository with user ada2015 with read permission.
- You have to login to the judge system (http://ada2015.csie.org) to bind the bitbucket repository to your account. For programming problems (those containing "**Programming**" in the problem topic), push your source code to your bitbucket repository. After you push the source code to repository, you can run the judge and get the score. Please refer to the guide of the judge system on its index page for further details.
- For other problems (also known as "hand-written problems"), submit the answers in an electronic copy via the git repository before the due time. Please combine the solutions of all these problems into **only ONE file** in the PDF format, with the file name in the format of "hw[# of HW].pdf" (e.g. "hw1.pdf"), all in **lowercase**; otherwise, you might only get the score of one of the files (the one that the grading TA chooses) or receive penalty because of incorrect filename.
- 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 (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 for problems with no specified references.
- **NO LATE SUBMISSION ONE DAY AFTER THE DUE TIME IS ALLOWED.** For all submissions, up to one day of delay is allowed; however, penalty will be applied to the score according to the following rule (the time will be in seconds):

$$\text{LATE\_SCORE} = \text{ORIGINAL\_SCORE} \times (1 - \text{DELAY\_TIME}/86400)$$

Note that late submission of partial homework is NOT allowed. The penalty will be applied to the entire homework in the case of late submission.

Problem 1

Fishy owns a stand in the fish market for many years. Fishy's fishes are very fresh and therefore the stand is always the most popular one in the fish market. Each fish takes Fishy 1 second to clean, and only cleaned fishes can be put onto the stand. Because of their popularity, Fishy's fishes will always be bought immediately by customers after staying for s seconds on the stand.

Today is Fishy's birthday so he decides to sell as few fishes as he can while satisfying the following constraint: he makes a list $t_1, t_2, ..., t_n$ where $t_1 < t_2 < \cdots t_n$ and insists to have exactly f fishes on the stand at time t_i second of the day in the list. Note that because today is Fishy's birthday so that he only sells, counts and starts to clean the fish on integer seconds, which means Fishy wouldn't take any action on 4.5 second, 4.55 second, etc.

Given a list $t_1, t_2, ..., t_n$ and constants s and f, Fishy will be really sad if he cannot satisfy the above constraint.

For example, if $s = 4$, $f = 2$ and the list is 5, 6, 11. Then a possible solution is listed below

t	1	2	3	4	5	6	7	8	9	10	11	12	13
Required f fishes	0	0	0	0	2	2	0	0	0	0	2	0	0
Start to clean fish	F	T	T	F	F	F	F	T	T	F	F	F	F
Fishes on stand	0	0	1	2	2	2	1	0	1	2	2	2	1

Fishy may start to clean a fish on $t = 2$, $t = 3$, $t = 8$ and $t = 9$ so that on $t = 5$, $t = 6$ and $t = 11$ there will be f fishes on the stand. A wonderful birthday has passed and Fishy is really happy that he only has to clean four fishes on his birthday, which is the minimum number of fish to fulfill the list.

(1) (4%) At which condition will Fishy be sad with his list no matter when he cleans the fishes? Express the condition using t_i, s, and f.

(2) (8%) If we know $f = 1$ and Fishy is not sad, can you design an algorithm so that Fishy can clean as few fishes as he can? Write an algorithm and prove its correctness.

(3) (8%) If we know Fishy is not sad, can you design an algorithm so that Fishy can clean as few fishes as he can? Write an algorithm and prove its correctness.

Problem 2

Do you know the famous game, League of Legends (LOL)? Recently, the fifth season of the world championship was held in Europe.

In LOL, each player controls a character and can buy equipments to increase the *influence* of the character on the match. It is a difficult problem to decide *when* and *in what order* to purchase a list of desired equipments. Let us model the problem more concretely, then we can try to solve it.

Consider that the player wants to get N equipments and each equipment has a price p_i dollars and an influence level b_i. In the game, the player can earn 1 dollar every second. The influence of a character at time t is simply the sum of the influence level of all the equipments the character owns at that time. That is, assume the player has a set of equipments E at time t, the influence is $\sum_{i \in E} b_i$. Then, our goal is to maximize the average influence at the moment when the player collects every equipment he wants. Let E_t be the equipment set that the player has at the second t, and T be the time (in seconds) the player needs to buy all equipments (i.e. $T = \sum p_i$). We want to find out the best buying order which maximizes the *average influence*: $\frac{\sum_{1 \le t \le T} \sum_{i \in E_t} b_i}{T}$.

In the following problems, we assume there are N equipments and all p_i, b_i are positive.

(1) (3%) Suppose now the player purchases equipments starting from the first one, then the second one, the third one, ...etc. Please write down the formula of our goal (average influence) using N, p_i, b_i, T $(1 \le i \le N)$.

(2) (3%) If the player decides to buy the $i+1$-th equipment first and the i-th later. The remaining part ($1 \dots i-1$ and $i+2 \dots n$) stays unchanged. Give a formula representing the change of the average influence. Your formula must only contain $p_i, p_{i+1}, b_i, b_{i+1}, T$

(3) (8%) Assume $f(x)$ is a function that calculates the value we want to maximize. In this problem, $f(S)$ is the average influence when buying order is S.

Given a sequence A, if for every $i < j$, there does not exist a sequence S s.t. $S_k = A_i$, $S_{k+1} = A_j$, and $f(S_{swap(k,k+1)}) > f(S)$, where S_k is the k-th item in S, $S_{swap(k,k+1)}$ is the sequence formed by swapping S_k and S_{k+1} of S. Then A is an optimal solution.

(4) (8%) Give an algorithm (in description or in pseudocode) to find a buying order that maximizes the average influence, and prove the correctness and the time complexity is $O(n \lg n)$.

(5) (3%) If p_i and b_i are integers, can you modify your algorithm to find out the order without using any float point calculation? Assume that all values are stored in `int` in C/C++. That is, the value of p_i and b_i will be between 1 and $2^{31} - 1$.

Please briefly describe the detail of you algorithm and assume that it is implemented under C/C++. You will lose some points if there exists any potential issue/bug.

Problem 3

Paul is playing a card game with his friend John. There are N piles of cards on the table. And there is a positive integer number on each card.

The players take turns and Paul takes the first turn. In Paul's turn he takes a card from the *top* of any non-empty pile, and in John's turn he takes a card from the *bottom* of any non-empty pile. Each player wants to *maximize* the total sum of the number on the cards he took. The game ends when all piles become empty.

There are N piles on the table, the i-th pile has n_i cards ($n_i \geq 1$). The integer number on the j-th card from the top of pile i is a_{ij}.

Suppose Paul and John both play optimally, what score will they get at the end of the game?

(1) (5%) Suppose $\forall i, n_i = 1$. That is, each and every pile has only one card with integer a_{i1}.

Prove that Paul has a strategy to get at least a score of

$$\sum_{j\, is\, odd} a_{[j]1},$$

and John has a strategy to get at least a score of

$$\sum_{j\, is\, even} a_{[j]1}.$$

Note that $a_{[j]1}$ denotes the j-th largest a_{i1}.

(2) (5%) Suppose $\forall i, n_i$ is even.

Prove that Paul has a strategy to get at least a score of

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i/2} a_{ij},$$

and John has a strategy to get at least a score of

$$\sum_{i=1}^{N} \sum_{j=n_i/2+1}^{n_i} a_{ij}.$$

(3) (7%) Suppose $\forall i, n_i$ is odd.

Prove that Paul has a strategy to get at least a score of

$$\sum_{i=1}^{N} \sum_{j=1}^{\frac{n_i-1}{2}} a_{ij} + \sum_{k\, is\, odd} a_{[k]\frac{n_i+1}{2}},$$

and John has a strategy to get at least a score of

$$\sum_{i=1}^{N} \sum_{j=(n_i+3)/2}^{n_i} a_{ij} + \sum_{k\, is\, even} a_{[k]\frac{n_i+1}{2}}.$$

Note that $a_{[k]\frac{n_i+1}{2}}$ denotes the k-th largest $a_{i\frac{n_i+1}{2}}$.

(4) (8%) Let us come back to the original problem.

Suppose Paul and John both play optimally, what score will they get at the end of the game? Note that n_i does not have any constraint now.

Hint: If $x \geq A$ and $y \geq B$ and $A + B = T = x + y$, then $x = A$ and $y = B$.

Problem 4 - Rabbit House (Programming)

Description

(30%) Kokoa is a coffee shop clerk of Rabbit House. Since Rabbit House is a famous coffee shop, it earns lots of money per day. Kokoa records the total income s_i everyday, and checks whether it matches the cash Rabbit House has now. To prevent malicious modification, she also records c_i, which is the checksum of s_i.

As a normal high school girl, she doesn't know the complicated message digest algorithms like MD5 or SHA-1. Instead, she just calculates the digit sums of s_i. The digit sum of a given integer is the sum of all its digits. For example, if $s = [302, 1000, 2011]$, the checksums will be $c = [5, 1, 4]$.

But this algorithm is too naive. It's easy to construct a different s with same checksums. For instance, if $s = [5, 10, 13]$, the checksums will also be $c = [5, 1, 4]$. To show how unreliable is digit sum, please construct a valid s for a given c, such that the last element of s is as small as possible. If there are more than one possible s, choose the one with smallest s_{n-1}, where n is the length of s. If there are still more than one possible s, choose the one with smallest s_{n-2}, and so on.

Input Format

The first line contains an integer T indicating the total number of test cases. Each test case starts with an integer n in one line, indicating the number of days, followed by a line of n integers $c_1, c_2, \ldots, c_n$.

- $1 \leq T \leq 10$
- $1 \leq n, c_i \leq 1000$

Output Format

For each test case, please output n integers $s_1, s_2, \ldots, s_n$ in one line. Note that s should be strictly increasing since Rabbit House is famous and has income everyday.

Sample Input

```
2
3
5 1 4
10
3 1 4 1 5 9 2 6 5 3
```

Sample Output

```
5 10 13
3 10 13 100 104 108 110 114 122 201
```

Hint

- How small could s_i be?
- How large could s_i be?