Assignment-4

Data Structures and Algorithms
Sorting

Due Date: 23 May, 2023 11:59 PM

Important Notes

- Doubt Document
- Plagiarism Policy. Please read this before you start the assignment
- Write your own sorting routines. (Taking routines from other sources is strictly prohibited and will result in a very heavy penalty)

TOTAL: TBD

1 To re-watch or not to re-watch

1.1 Problem description

There are N movies on your current streaming subscription and watching the i^{th} one for the first time awards you with A_i units of happiness. However, once you watch a movie, its entertainment value gets decremented by 1.
That is, formally, if the current entertainment score of the i^{th} movie is A_i and you decide to watch it, then, your happiness increases by A_i and A_i is decremented by 1.

For example, if your current happiness is say 5 and the current entertainment scores of the movies are $[2, 3, 10, 0]$, then, watching the 2nd movie say, results in you having a happiness of 8 and the new entertainment scores as $[2, 2, 10, 0]$.

You start off with a happiness score of 0. You want to watch **at most** k movies in total and can watch any movie any number of times. **Note that re-watching a movie also counts towards this total.** What is the **maximum happiness score** that you can achieve?

1.2 Constraints

- All input values are integers
- $1 \leq N, K \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^9$
- $1 \leq T \leq 2 \times 10^5$
- Sum over N, K across all test-cases does not exceed 2×10^5

1.3 Input format

The first line of input contains a single integer T that denotes the number of test-cases. Then, $2T$ lines follow.

The first line of each test-case contains two space-separated integers N, K that denotes the number of movies and the limit on number of movies that can be watched.
The second line of each test-case contains N space-separated integers $A_1, A_2, \ldots, A_N$ that denotes the initial entertainment scores of the N movies.

That is, it is in the following format

```
T
N K
A_1 A_2 ... A_N
...
```

1.4 Output format

For each test-case, output on a single line the maximum happiness score that can be achieved.

1.5 Example test-case

Input	Output
2 4 2 2 3 10 0 1 4 5	19 14

Input	Output
1 5 4 1 3 2 1 5	15

2 Magic Marbles

2.1 Problem description

There are N marbles lying on the floor with each of them having an integer A_i written on them. For each marble, you can either leave the number on it untouched, or replace it with a pre-determined integer B_i. You need to **pick up all the marbles** but you may do so in **any order**.

You travel with two journals J_1 and J_2. Whenever you pick up a marble, regardless of the current number on it, you append an entry in J_1 with the **original integer** A_i corresponding to that marble. Similarly, you append the actual value written on the marble (either A_i or B_i depending on whether or not you replaced the number on it) to J_2.

You love sorted arrays and so, would like to keep your journals sorted in a **non-decreasing** fashion. What is the **minimum value** that the **last entry** (that is, the entry corresponding to the last marble picked up) in J_1 and J_2 can take? It can be shown that it is always possible to make both the journals sorted.

2.2 Constraints

- All input values are integers
- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A_i, B_i \leq 10^9$
- $1 \leq T \leq 2 \times 10^5$
- Sum over N across all test-cases does not exceed 2×10^5

2.3 Input format

The first line of input contains a single integer T that denotes the number of test-cases. Then, $3T$ lines follow.

The first line of each test-case contains a single integer N that denotes the number of marbles.
The second line of each test-case contains N space-separated integers $A_1, A_2, \ldots, A_N$ that denotes the initial numbers written on the marbles.
The third line of each test-case contains N space-separated integers $B_1, B_2, \ldots, B_N$ that denotes the alternate numbers that you may replace the original integers with. That is, B_i may replace the value currently written on the i^{th} marble. Note that this does not affect the entry appended to J_1 when the i^{th} marble is picked.

That is, it is in the following format

```
T
N
A_1 A_2 ... A_N
B_1 B_2 ... B_N
...
```

2.4 Output format

For each test-case, output on a single line, output two space-separated integers, the minimum value that the last entry in J_1 and J_2 can take respectively.

2.5 Example test-case

Input	Output
1 5 1 4 10 3 11 2 1 6 4 8	11 8

Input	Output
2 3 1 1 1 1 1 1 4 1 2 3 4 1 2 1 1	1 1 4 4

3 Radical Sorting

3.1 Problem description

You have N strings that you want to sort Radically.

String A occurs **before** a string B in Radically sorted order if and only if at the first position of difference from left, the following holds:

- it is an odd position and A has a letter that appears earlier in the alphabet than the corresponding letter in B
- it is an even position and A has a letter that appears later in the alphabet than the corresponding letter in B
- A is a prefix of B and A has an odd length
- B is a prefix of A and B has an even length

For example, consider two strings *cbfz* and *cefb*. The first point of difference from left is at index 2. As 2 is **even** and e comes **after** b in the alphabet, therefore *cefb* comes **before** *cbfz* in a radically sorted order.

Given N strings, sort them Radically.

NOTE: 1-based indexing is followed to determine if the position is odd or even.
NOTE: <**string.h**> is **not allowed** in this question.

3.2 Constraints

- $1 \leq T \leq 2 \times 10^5$
- $1 \leq N \leq 2 \times 10^5$
- Sum of $N * M$ over all test-cases does not exceed 6×10^5
- N is the number of strings
- M is the maximum length of the string
- All the strings contain **only** lowercase english alphabet

3.3 Input format

The first line of input contains a single integer T that denotes the number of test-cases. Then the description for each test-case is as follows.

The first line of each test-case contains a single integer N that denotes the number of strings.
Then N lines follow for each test-case, the i^{th} of which contains an integer m, followed by a string of length m.

That is, it is in the following format

```
T
N
m_1 string_1
m_2 string_2
...
```

3.4 Output format

For each test-case, output the radically sorted strings, each string on a new line.

3.5 Example test-case

Input	Output
2 3 3 abc 4 azcd 4 abcf 5 3 bdf 2 zz 1 z 3 baa 3 abd	azcd abc abcf abd bdf baa z zz

4 The Schoolyard Shuffle

4.1 Problem Description

K schools meet together every year to host the Schoolyard Shuffle. Each school sends M students. Each student has a unique combat power. The combat powers of $K \cdot M$ students from all schools are given in an array A, arranged in a specific pattern.

$$\begin{array}{ll} \text{Students of School 1:} & A[1], \quad A[K+1], \quad A[2K+1], \quad \ldots, \quad A[(M-1)K+1] \\ \text{Students of School 2:} & A[2], \quad A[K+2], \quad A[2K+2], \quad \ldots, \quad A[(M-1)K+2] \\ \vdots & \\ \text{Students of School K:} & A[K], \quad A[2K], \quad A[3K], \quad \ldots, \quad A[M \cdot K] \end{array}$$

The first K students fight against each other, and the one with the maximum combat power among them wins. This is followed by the next K students, and so on. Each student can fight only once.

A random school J is chosen each year which is given the special privilege of shuffling the order of it's students in any order at most once. Find the maximum number of wins school J can get at the end of the competition.

NOTE: 1-based indexing is followed in the problem description.

4.2 Constraints

- $1 \leq T \leq 2 \times 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq J \leq K$
- $1 \leq A[i] \leq 10^9 \ \forall \ i \in [1, K \cdot M]$
- Sum of $K * M$ over all test-cases does not exceed 10^6
- K is the number of participating schools
- M is the number of students from each school
- J is the chosen school that is allowed to shuffle it's students

4.3 Input Format

The first line of input contains a single integer T that denotes the number of test-cases. Then $3T$ lines follow.

The first line of each test-case consists of 2 space separated integers K and M.
The second line of each test-case contains $K \cdot M$ space-separated integers $A[1], A[2], ..., A[K \cdot M]$ that denotes the combat powers of each participating student. Each combat power is unique.
The third line of each test-case contains an integer J, denoting the chosen school.

That is, it is in the following format

```
T
K M
A_[1] A_[2] ... A_[K*M]
J
```

4.4 Output Format

For each test-case, output a single integer representing the maximum number of wins the school J can get.

4.5 Example Test case

Input	Output
1 3 5 8 5 3 14 9 22 29 13 2 36 47 17 50 51 11 2	3

Explanation:
Here, in the first test-case

Students of School 1: 8, 14, 29, 36, 50
Students of School 2: 9, 51, 5, 47, 13
Students of School 3: 3, 22, 2, 17, 11

is one of the optimal arrangements where the 2^{nd} school has the privilege to shuffle it's students which gives an output 3 by obtaining a win in the first, second and fourth fights.