TeamByte

IEEEXtreme 10.0 > P = NP?

P = NP?

locked

by *IEEEXtreme*

Problem Submissions Leaderboard Discussions

When most computer scientists ask, "Does P = NP?", they are asking whether there are decision problems that can be verified in polynomial time, but not solved in polynomial time.

However, when Dr. I.M. Columsie asks this question, he is wondering about the relationship between pizza consumption and success in programming contests. His current research analyzes the results of a previous Xtreme contest. He gathered the final rankings for a selected subset of teams, and divided them into two, non-empty groups - a pizza (*p*) group, where team members ate copious amounts of pizza during the contest, and a no pizza (*np*) group, where the team members did not eat pizza. His hypothesis is that the pizza improves performance in these contests.

He wanted to compare the rankings of the two groups to see if the data supports his hypothesis. He started by writing down the sorted rankings within the *p* group and the *np* group. He noted that there were no teams that tied in the data he gathered, so all of the rankings were unique (both within each group and across the two groups).

He was about to compute the mean of the rankings for the two groups, when he spilled coffee all over his computer and the written sorted rankings. Unfortunately he lost the raw data on the computer, and a number of the written rankings were no longer readable.

He really hopes that you will be willing to help him by calculating a range of possible values for the sums of the rankings for the two groups. From these sums, he will be able to calculate a range of possible values for the means.

Input Format

The input starts with three integers ***t***, ***p***, and ***n***, where ***t*** gives the number of teams in the contest, ***p*** gives the size of the pizza group, and ***n*** gives the size of the ***no pizza*** group.

Then there is a blank line.

The next ***p*** lines give the rankings, sorted in ascending order, for the ***pizza*** group. If a ranking is unknown, it will be listed as `?`.

After this first group of rankings, there is a blank line.

Following this are ***n*** lines that similarly give the rankings for the ***no pizza*** group.

Constraints

$1 \leq n + p \leq t <= 2 * 10^5$

Each ranking must be between 1 and ***t***, inclusive.

Output Format

The output consists of two lines in the following format:

```
[p_sum_low] [p_sum_high]
[np_sum_low] [np_sum_high]
```

where

- `[p_sum_low]` and `[p_sum_high]` are the lowest and highest possible values for the sums of the rankings of the ***p*** group
- `[np_sum_low]` and `[np_sum_high]` are the lowest and highest possible values for the sums of the rankings of the ***np*** group

Note that it is guaranteed that there is at least one ranking that fits the data.

Sample Input

```
100 5 6

1
?
?
7
```

```
96

?
?
4
?
98
?
```

Sample Output

```
115 115
214 304
```

Explanation

Because the rankings must be unique, we can deduce constraints on the missing values. For example, since the `1` ranking is in the *p* group and the first two rankings in the *np* group must be less than `4`, the first two missing values in the *np* group must be `2` and `3`. We can also conclude that the first two missing values in the *p* group must be `5` and `6`. For the remaining missing values, we can establish multiple possibilities.

We know every value from the *p* group: [1, 5, 6, 7, 96], and, thus, we can establish an exact value for the sums of the possible rankings of this group.

The *np* group consists of the following list of values: [2, 3, 4, *a*, 98, *b*] where *a* is a value chosen from the set {8, 9, ..., 95, 97}, and *b* is a value chosen from the set {99, 100}. The smallest sum of the rankings would then be 214, if the rankings were [2, 3, 4, 8, 98, 99], and the largest sum of the rankings would be 304, if the rankings were [2, 3, 4, 97, 98, 100].

Max Score: 95pts dynamic

Submissions: 120

Max Score: 95

Difficulty: Hard

More

Current Buffer (saved locally, editable)

C++

```
#include <cmath>
#include <cstdio>
```

```
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Scoring | Environment | FAQ | Terms Of Service | Privacy Policy