The Protest Contest

USC Programming Contest, Spring 2018

April 01, 2018

Sponsored by VSoE, Electronic Arts, Google, Northrop Grumman

The last few months have seen increased protests from students (and many other supporters), galvanized by recent mass shootings at schools, in particular in Parkland, FL. Millions of young people are finding a public voice and interest in (and rage at) politics. Somewhat coincidentally, this is also the 50$^{\text{th}}$ anniversary of 1968, which is remembered (in the US, and worldwide) for extensive protests, also often led by students. Those protests were in response to a number of grievances, particularly the Vietnam War, widespread societal racism, and in particular racial and other inequalities in access to and quality of education. In this contest, we will explore a number of computational problems relating to protests and social engagement.

You can solve or submit the problems in any order you want. When you submit a problem, you submit your source code file using PC^2. Make sure to follow the naming conventions for your source code and the input file you read. Remember that all input must be read from the file of the given name, and all output written to `stdout` (the screen). You may not use any electronic devices besides the lab computer and may not open any web browser for whatever reason.

One thing you should know about this contest is that the judging is done essentially by a `diff` of the files. That means that it is really important that your output follow the format we describe — if you have the wrong number of spaces or such, an otherwise correct solution may be judged incorrect. Consider yourselves warned!

Another warning that seems to be in place according to our experience: all of our numbering (input data sets, people, etc.) always starts at 1, not at 0.

And a piece of advice: you will need to print floating point numbers rounded to two decimals. Here is how you do that:

C: `printf ("%.2f", r);`.

C++: `cout.precision (2); cout << fixed << r;`.

Java: `System.out.print ((new java.text.DecimalFormat("#.##")).format(r));`

[Including this page, the problem set should contain 8 pages. If yours doesn't, please contact one of the helpers immediately.]

Problem A: Tide Pods

File Name: `tidepods.cpp|tidepods.java`

Input File: `tidepods.in`

Description

Many conservative commentators try to discredit the anti-gun protest movement on the grounds that we shouldn't listen to members of a generation that eats tide pods.[1] The main problem with such an attack is that just because someone belongs to a group (here: American teenagers) some of whom do stupid things doesn't mean that the individual does the same stupid things. Attacking someone based on membership in a large group typically reveals that one is not willing or able to address the actual opinion or message. Indeed, the "tide pod" argument can easily be turned around: if you are a female American teenager, you share being an American teenager with some stupid guy eating a tide pod, but you also share being female and under 21 with Malala Yousafzai. In this problem, you will compute the largest difference in opinion one could form about individuals by likening them to the best and worst other people they may resemble.

More precisely, you will be given a list of (binary) traits (e.g., being a teenager, being American, being female). There is a list of model people to compare an individual to: for each, you will be given which traits they possess, and a score of how "great" a person they are. The individual's similarity to a model person is the number of traits they have in common. (Similarity is not acquired by missing the same trait: that is, we don't say that Malala Yousafzai and Neil deGrasse Tyson have in common that they are *not Swiss*.) The comparison score is then the product of the number of traits they share with the score of the model person. You are to print the difference between the largest and smallest comparison score.

Input

The first line contains a number $K \geq 1$, which is the number of input data sets in the file. This is followed by K data sets of the following form:

The first line of each data set contains two integers t, n. $1 \leq t \leq 10$ is the number of traits we use for comparing people, and $1 \leq n \leq 1000$ is the number of model people.

This is followed by n lines, each describing one model person i. Each such line first contains t numbers $a_{i,j} \in \{0, 1\}$, where $a_{i,j} = 1$ means that person i has trait j. The last number on this line is the person's integer score s_i with $-10000 \leq s_i \leq 10000$.

Finally, there is one line describing the individual to evaluate, containing t numbers $b_j \in \{0, 1\}$, which capture the traits that the individual has.

Output

For each data set, first output "`Data Set x:`" on a line by itself, where `x` is its number. Then, output the maximum difference in scores for the individual.

Each data set should be followed by a blank line.

Sample Input/Output

Sample input `tidepods.in`

```
2
4 3
1  0 0 0  -5
1 1  0 1  2
0 1  1 0 3
1 0 1  1
2 3
1 0 -100
0 0 101
0 1 -1
0 1
```

Corresponding output

```
Data Set 1:
9

Data Set 2:
1
```

[1] This refers to a few teenagers who posted video of themselves eating tide pods (small colorful plastic containers of laundry detergent).

Problem B: Blowin' in the Wind

File Name: `blowin.cpp|blowin.java`

Input File: `blowin.in`

Description

An important part of the 1968 protest movement were protest songs: songs with socially conscious or protest lyrics, often set to simple folk-like music. One of the iconic songs was "Blowin' in the Wind" by Bob Dylan[2], which starts with the lyrics "How many roads must a man walk down before you can call him a man?" Obviously, Dylan did not mean to imply that a certain minimum number of walked roads is the criterion for manhood — rather, the roads metaphorically represent the life experiences one has. In this problem, you will compute the minimum number of roads to walk to attain all requisite life experiences in the correct order.

In this problem, you will be given some life goals to fulfill in the given order. You will also be given a (connected, undirected) graph. For each life goal, there will be at least one node of the graph at which the life goal can be fulfilled. You are to compute the minimum number of roads (edges) you need to traverse from your start node (always node 1) until you have fulfilled all life goals.

Input

The first line contains a number $K \geq 1$, which is the number of input data sets in the file. This is followed by K data sets of the following form:

The first line of the data set contains two integers $1 \leq g \leq 20$ and $1 \leq n \leq 100$. g is the number of life goals, and n the number of nodes in the graph.

This is followed by n lines, each line $i = 1, 2, \ldots, n$ describing a node i. Each line begins with g integers $a_{i,j} \in \{0, 1\}$, where $a_{i,j} = 1$ means that life goal j can be attained at node i. The remaining entries of the line are integers in $\{1, 2, \ldots, n\}$ and describe edges for node i. Edges are undirected. The same edge may appear multiple times in the list, and when edge (i, j) is listed for node i, it may or may not also appear for node j — either way, it can be used to get from node j to node i.

Output

For each data set, first output "`Data Set x:`" on a line by itself, where `x` is its number. Then, output the minimum number of roads you must walk down, starting from node 1, to meet all life goals in the order $1, 2, 3, \ldots, g$.

Each data set should be followed by a blank line.

Sample Input/Output

Sample input `blowin.in`

```
1
4 8
0 0 1 0 2 4
1 0 0 0 3
0 0 0 0 6 2
0 1 0 1 5 1
0 0 0 0 7 8
0 0 0 0 7
0 1 1 0 8 5
1 0 0 1
```

Corresponding output

```
Data Set 1:
5
```

[2]Recipient of the 2016 Nobel Prize for literature.

Problem C: Racial Discrimination

File Name: `racial.cpp|racial.java`

Input File: `racial.in`

Description

One of the main student protests of 1968 was at San Fernando Valley State College (now Cal State Northridge), where many African American students protested against the unequal treatment of minority students. Racial, gender, and other discrimination are still widespread today. Two problems in countering them are that (1) it can be easy to discriminate accidentally, and (2) as a result, it can be hard to prove that a person or institution discriminated intentionally. As algorithms and machine learning techniques are used increasingly frequently to make or guide important decisions, it is imperative to understand precisely what we even mean by discrimination, and how to counteract it.

An easy way to suggest that intentional discrimination may have happened is to show some protected categories such that the decisions align (almost) perfectly with a division by those categories. For instance, if the data show that all merit fellowship applications were rejected except those by Caucasian men, this would make discrimination likely. On the other hand, to defend oneself from charges of discrimination, one could point to *non-protected* categories which (almost) perfectly explain decisions. For example, if merit fellowships were given exactly to students with a GPA of at least 3.8 and membership in at least 3 campus organizations, then this would suggest no illegal choices.

Here, you will compute how easily your observed data can be explained by just looking at some small number c of categories. For each of n individuals and m categories, you will be given whether the individual belongs to the category, as well as whether the individual was selected for a fellowship. Then, you are to decide what is the minimum number of individuals you can misclassify if you only get to look at c of the categories. Notice that you can use any rule based on those c categories. For example, if all Caucasian male and all African-American female applicants got a fellowship, and no African-American male and only one Caucasian female, then you would only miscategorize one individual (the Caucasian female) with the rule "Caucasian male, African-American female, no one else." In other words, if you can pick any c (out of m) categories, and form any "fellowship" rule using arbitrary combinations of those c categories, what is the minimum number of students whose fellowship status you will get wrong by using the best such rule?

Input

The first line contains a number $K \geq 1$, which is the number of input data sets in the file. This is followed by K data sets of the following form:

The first line of a data set contains three integers n, m, c. $1 \leq n \leq 200$ is the number of individuals, $1 \leq m \leq 10$ is the number of categories, and $0 \leq c \leq m$ is the number of categories you can use in your rule.

This is followed by n lines, each containing $m + 1$ bits. The first m bits $a_{i,j}$ tell you for each of the m categories whether i belongs to category j, which is encoded by $a_{i,j} = 1$. The last bit tells you whether individual i was selected for a fellowship.

Output

For each data set, first output "`Data Set x:`" on a line by itself, where `x` is its number. Then, output the minimum number of individuals who can be miscategorized by any rule using only c categories.

Sample Input/Output

Sample input `racial.in`

```
1
7 4 2
0 0 0 0 1
0 0 0 1 0
1 1 1 0 0
1 1 0 1 1
1 0 1 0 1
0 1 0 0 0
1 1 1 1 0
```

Corresponding output

```
Data Set 1:
1
```

Problem D: Non-Violent Protests

File Name: `riot.cpp|riot.java`

Input File: `riot.in`

Description

Another historic event in 1968 was the assassination of Martin Luther King Jr. He was a leader of the Civil Rights Movement, and his "I have a dream" speech is considered one of the most important speeches in American history. Martin Luther King Jr. was an unwavering advocate for non-violent protests. The reason non-violent protests tend to be more effective is that the goal of protests is typically to raise awareness and sway public opinion. It is much harder to sway public opinion in your favor when you are perceived as violent. Not surprisingly, then, police and other organizations tried to undermine the Civil Rights movement by aggressive provocation, as well as by planting violent protesters in the hopes that they would lead entire protests to follow their example and start large-scale riots. One important point that this raises is that for a protest in a tense environment, the line between a peaceful protest and a riot is often very thin. While reports in the media (and in history books) will typically treat the character of the whole group as "uniform," in reality, these are individuals reacting to a particular situation, and very subtle changes in the situation may lead to different outcomes.

We will explore this with a simplified model, as follows. n individuals are participating in a protest. Each individual has an integer threshold $t_i \geq 0$. Individual i will riot if at least t_i other people are already rioting before him. In turn, he may thus cause others to riot, too. Starting from no one rioting, you are to compute how many people will riot at the end.

Input

The first line is the number K of input data sets, followed by K data sets, each of the following form:

The first line is the number $0 \leq n \leq 1000000$ of individuals. This is followed by a line with the n thresholds t_i of the individuals, given in no particular order.

Output

For each data set, output "`Data Set x:`" on a line by itself, where `x` is its number. Then, output the number of people who riot at the end.

Each data set should be followed by a blank line.

Sample Input/Output

Sample input `riot.in`

```
2
6
1 1   1  1  1   1
8
3 0 0  10 4  6 6   2
```

Corresponding output

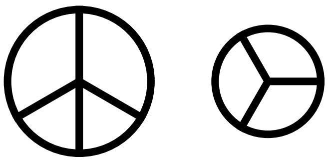
```
Data Set 1:
0

Data Set 2:
5

```

Problem E: Peace Sign

File Name: `peace.cpp|peace.java`

Input File: `peace.in`

DescriptionOne of the iconic signs of the 1960s protest movement was the peace sign, shown in the picture below on the left side. On the right, you see (a rotated, scaled version of) the sign for Mercedes Benz, a manufacturer of luxury cars. The similarity between the two has been noted often, and some of the protesters in the 1968 movements have been accused of trading the left sign for the right in their own lives.



Without analyzing whether former protesters ended up being more economically successful — or less socially progressive — later in life than their contemporaries, we will instead focus here on the much more important question how to determine how similar two line drawings are. You will be given two drawings, each consisting of some number of straight lines (no circles). The similarity is the total number of lines in the second image you can exactly match by taking the first image, and doing a combination of (1) translating it some combination of up/down/right/left, (2) rotating it by some angle, and (3) scaling it by some factor. For example, in the given picture (without the circle), you can rotate the left picture 90 degrees to the right, shift it to the right, and scale it by 75%, and match three lines.

Input

The first line is the number K of input data sets, followed by the K data sets, each of the following form:

The first line of the data set contains two integers $1 \leq n, n' \leq 50$, the number of lines in the first and second image. This is followed by a line with $4n$ doubles $0.0 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 100.0$; the i^{th} line in the first image goes from $(x_{i,1}, y_{i,1})$ to $(x_{i,2}, y_{i,2})$. Next is a line with $4n'$ doubles $0.0 \leq x'_{i,1}, y'_{i,1}, x'_{i,2}, y'_{i,2} \leq 100.0$; these describe the n' lines of the second image in the same way.

Output

For each data set, output "`Data Set x:`" on a line by itself, where `x` is its number. Then output the maximum number of lines in the second image you can match by translating, rotating, and scaling the first image. Our inputs will be such that tiny rounding errors of less than 0.00001 will not affect the results.

Each data set should be followed by a blank line.

Sample Input/Output

Sample input `peace.in`

```
4
4 3
3 3  3 1   3 3  3 5   3 3  4.73205 2   3 3  1.26795 2
2.5 2.5  4 2.5   2.5 2.5  1.75 3.79904  2.5 2.5  1.75 1.20096
3 3
0 1  3 1   0 2  2 2  0 0  1 0
0 1  3 1   0 2  1 2  0 0  2 0
3 3
0 1  3 1   0.5 2  2.5 2  1 0  2 0
0 1  3 1   1 2  2 2  0.5 0  2.8 0
2 4
0 1  2 1   1 0  1 2
0 1  1 1   1 1  2 1   1 0  1 1   1 1  1 2
```

Corresponding output

```
Data Set 1:
3

Data Set 2:
1

Data Set 3:
2

Data Set 4:
1
```

Problem F: Gun Control

File Name: `gun.cpp|gun.java`

Input File: `gun.in`

Description

The topic that is currently galvanizing a new generation of students to protest, and hopefully to engage in the political process beyond that, is gun control. Even though a vast majority of Americans favor stricter gun control than the status quo, and despite horrible gun death statistics[3], gun laws have if anything been weakened over the past 2–3 decades. One of the main reasons is the disproportionately large influence of the National Rifle Association (NRA). In the past, the NRA mostly represented gun owners, but at this point, it is de facto a lobbying arm for gun manufacturers. It promotes essentially the following reasoning: (1) The second amendment is absolute, and no law should restrict gun ownership. (2) In particular, our member companies have every right to sell guns to criminals and unstable people. (3) Hey, look at all those criminals with guns! You better buy a bunch of guns to protect yourself and your family from them.

Part of the reason the NRA has so much power over politicians is having created a situation where a large fraction of lawmakers (in particular, nearly all Republican lawmakers) is already advocating the NRA's position. This allows them to immediately target individual dissenters with demonizing ad campaigns. Given that they have the resources to unelect almost any individual lawmaker, no individual (or small coalition) dares change their position. If a large fraction of lawmakers were to simultaneously change their position, they would likely be protected by their number and the NRA's lack of resources to attack them. Unfortunately, such a large coalition likely would not be able to agree on a lot of gun control; each of them would have another little thing that would require a compromise. You are to compute what is the minimum total compromise that a coalition can accomplish such that at least one of its members is guaranteed to not be unelected.

More formally, for each of the lawmakers, you are given two numbers: the amount of money the NRA would have to spend to defeat this lawmaker, and the amount of compromise this lawmaker would require. A coalition S of lawmakers is *safe* if the combined amount of money required to defeat all of them exceeds the NRA's budget. The *compromise* required from S will be the sum of all their individual compromise levels. You are to find the least amount of compromise that can be achieved by any safe coalition.

Input

The first line contains a number $K \geq 1$, which is the number of input data sets in the file. This is followed by K data sets of the following form:

The first line of a data set contains two integers n, B. $1 \leq n \leq 50$ is the number of lawmakers, $0 \leq B \leq 1000$ is the NRA's budget.

This is followed by n lines, each containing two non-negative integers b_i, c_i. $b_i \leq 1001$ is the amount of money it would take the NRA to defeat lawmaker i, and $c_i \leq 1001$ is the amount of compromise that lawmaker i would require. We guarantee that $B < \sum_i b_i$, so there is a safe coalition.

Output

For each data set, first output "`Data Set x:`" on a line by itself, where `x` is its number. Then, on a line by itself, output the minimum amount of compromise that any safe coalition of lawmakers can achieve.

Each data set should be followed by a blank line.

Sample Input/Output

See next page.

[3] A large fraction are in fact suicides.

Sample input `gun.in`

```
1
8 10
1000 20
2 2
3 4
4 4
2 4
5 6
6 7
0 0
```

Corresponding output

```
Data Set 1:
12
```