Algorithms for Programming Contests WS20/21 - Week 07

Chair for Foundations of Software Reliability and Theoretical Computer Science, TU München

Mikhail Raskin, Martin Helfrich, Maxi Weininger, Christoph Welzel

Welcome to our practical course! This problem set is due by

Wednesday, 23.12,2020, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

This page is intentionally left blank.

Problem A
Street Lights

During another of Lea's visits to her uncle in Chaosville, she realizes that the streets of Chaosville are in horrible condition. Driving at night is really dangerous: you never know when you will hit a pothole, because most street lights are not working and you see close to nothing. One day, Lea decides to visit the mayor's office to talk about the bad lighting on the streets. The mayor explains the city's problem to Lea: indeed, there are many street lights, but erected in an irregular fashion at the side of every street. And most of them are switched off due to severe budget cuts. Not willing to give up, Lea offers to come up with a plan that specifies how many of the street lights should be switched on to illuminate the main street, at least. The mayor agrees, but only if she can provide him with a solution that needs to switch on only as few street lights as possible.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with three integers ℓ n d, the length of the main street ℓ, the number of street lights on main street n, and the radius of the light cone of each street light d, which indicates how far each light shines. Then, another line follows, consisting of n integers p_1 p_2 $\ldots$ p_n, describing the positions of the street lights.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either the smallest number of street lights that are needed to illuminate the whole main street (which goes in a straight line from 0 to ℓ), or "impossible" if there is no way to illuminate the whole street.

To illuminate the whole main street, there should be light from at least one street light at every point on the main street between 0 and ℓ. The boundary of each light cone is considered to be illuminated as well. In particular, this means that a point on the street is illuminated if two light cones touch there, they do not need to intersect. See the sample data explanation.

Constraints

- $1 \leq t \leq 20$
- $1 \leq \ell \leq 50000$
- $0 \leq n \leq \min(1000, \ell + 1)$
- $0 \leq d \leq 1000$
- $0 \leq p_i \leq \ell$ for all $1 \leq i \leq n$
- $p_i \neq p_j$ for all $1 \leq i, j \leq n$

Sample Data Explanation

In the first sample case, three street lights need to be switched on. Note that these may also be three different street lights, for example switching on the first, third, and sixth street light works as well. The second and third sample cases are impossible as there is always at least one section of the street that cannot be illuminated. In the fourth sample case, the street can be illuminated on its entire length by switching on all street lights. The points at which the light cones meet between the street lights are illuminated as well.

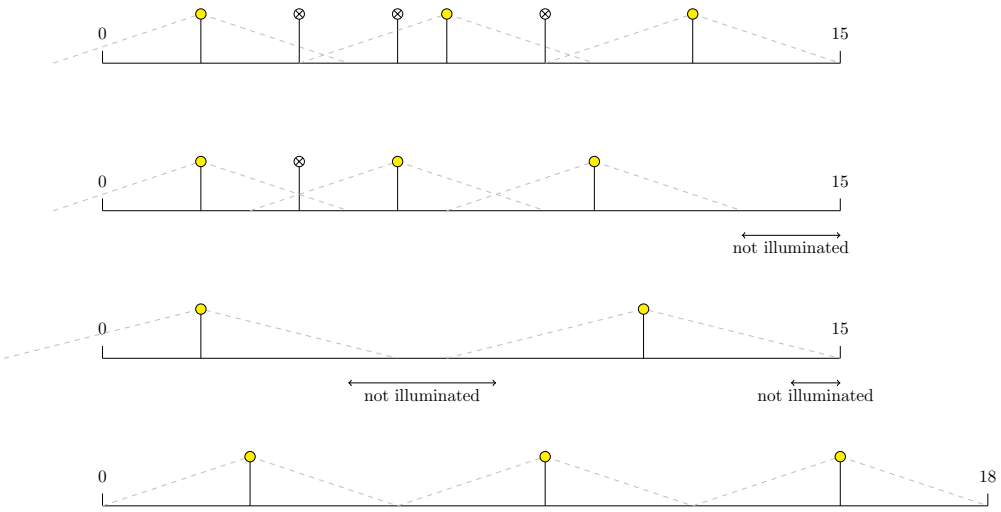

Figure A.1: Visualization of the first four sample cases.

Sample Input 1

```
2
15 6 3
2 4 9 7 6 12

15 4 3
10 4 6 2
```

Sample Output 1

```
Case #1: 3
Case #2: impossible
```

Sample Input 2

```
7
15 2 4
2 11

18 3 3
3 15 9

21 14 4
14 8 10 7 20 21 3 6 18 15 16 12 9 5

9 9 3
1 2 7 3 8 0 9 6 4

14 12 5
2 3 12 10 8 1 7 0 5 13 14 11

23 14 3
2 9 7 11 5 19 8 0 23 14 15 6 21 10

14 8 4
14 8 0 6 13 9 10 1
```

Sample Output 2

```
Case #1: impossible
Case #2: 3
Case #3: 3
Case #4: 2
Case #5: 2
Case #6: 5
Case #7: 3
```

Problem B
Soup Delivery

Lea has invented and patented Vegan Chicken Soup. The soup is biochemically identical to the ordinary chicken soup, but can be produced by genetically modified bacteria in a certified organic and vegan process using solar energy. As organic certification required to exclude any contamination risks, the bacteria are modified to require twice the normal atmospheric pressure to survive. While the process can be run with perfect safety in a cheap ordinary pressure cooker, some anti-progress rental contracts would prohibit such a setup. Lea has also invented a very cheap soup stasis pot keeping the soup perfectly fresh during delivery, but it is expensive to operate with the cost proportional to the time.

Multiple potential customers in Munich want perfectly fresh Vegan Chicken Soup delivered at exactly 10:00 every weekday. Of course, Lea doesn't want to turn any customers away. She has found a few spots suitable for the production points reasonably close to the customer locations. Now Lea needs to rent some of the available production locations to minimise the sum of rent and delivery expenses. Lea hosts her company's website on Amazon Web Services, so paying a few times more than strictly necessary doesn't shock her; still, the cheaper the better. Help Lea optimise the costs without sacrificing on quality!

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two space-separated integers N and M. Here N is the number of available production locations and M is the number of customers.

A line containing N space-separated integers $c_1, \ldots, c_N$ follows. The number c_i is the monthly cost of maintaining the i-th production location.

Afterwards N lines follow with the i-th line containing M space-separated integers $d_{i,1}, \ldots, d_{i,M}$ each. The number $d_{i,j}$ is the monthly cost of delivering the soup from i-th production location to the j-th customer.

It is guaranteed that the delivery costs are metric. In other words, one can additionally define delivery costs between the production locations and between the customer locations, such that the resulting pairwise delivery costs are non-negative, zero only for delivery from a point to itself, symmetrical, and satisfy triangle inequality.

Output

For each test case output a line containing "Case #i: v", where v is a 4-optimal value. k lines follow. The i-th line contains space-separated integers n_i, $m_{i,1}$, …, m_{i,M_i}, meaning that facility n_i serves customers $m_{i,1}$ to m_{i,M_i}. Each customer must be served by exactly one location, and the total cost of delivery and maintaining all the locations serving at least one customer must be within the factor of 4 to the minimum possible cost.

Constraints

- $1 \leq t \leq 20$
- $1 \leq N \leq 100$
- $1 \leq M \leq 200$
- $1 \leq c_i \leq 10^6$
- $1 \leq d_{i,j} \leq 10^6$

Sample Input 1

```
1
2 2
2 2
1 10
10 1
```

Sample Output 1

```
Case #1: 6
1 1
2 2
```

Sample Input 2

```
1
8 9
2 9 6 5 4 4 6 3
12 9 19 19 6 8 7 10 17
1 16 7 7 8 9 15 18 10
9 17 16 16 6 3 15 19 15
11 8 17 18 6 9 6 9 14
6 15 13 13 4 2 14 17 11
7 10 12 13 3 6 9 12 10
15 3 18 19 12 15 9 8 7
12 18 18 19 7 5 17 20 17
```

Sample Output 2

```
Case #1: 67
1 7
2 1 3 4
5 5 6
7 2 8 9
```

Problem C
Customs

Making money is hard, not just for people but also for countries. It is so hard that an especially broke country, Poorland, has come up with a completely new idea: They will split their country into two countries. This way, they can collect customs duties each time someone passes from one country to another.

As they want to maximize the amount of money they collect, the number of roads that lead from one of the new countries to the other should be maximal. Finding an optimal solution is very costly, thus they asked Lea to find a solution that differs from the optimal solution by at most a factor of $\frac{1}{2}$. As streets start and end at cities, the task will be to partition the cities into two sets.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a number n, the number of cities in Poorland, n lines follow. The i-th line describes city i and contains an integer k_i followed by k_i integers $c_1, ..., c_{k_i}$. k_i is the number of streets exiting city i, $c_1, ..., c_{k_i}$ are the cities directly connected to city i by streets. Cities are indexed from 1 to n.

You can assume that streets are undirected: If city i is directly connected to city j, then city j will be directly connected to city i.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. Output one more line containing integers $a_1...a_l$ in ascending order such that when the country is partitioned into the cities $a_1, ..., a_l$ and the remaining cities, the number of roads between the parts is at least half as big as in the optimal solution. If there are multiple solutions, any of them will be accepted. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 300$
- $0 \leq k_i \leq n$ for all $1 \leq i \leq n$
- $1 \leq c_i \leq n$ for all $1 \leq i \leq k_i$

Sample Input 1	Sample Output 1
2 2 1 2 1 1 4 1 3 1 4 2 1 4 2 2 3	Case #1: 1 Case #2: 1 2 4

Sample Input 2

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

4
0
2 3 4
1 2
1 2

5
1 4
0
0
1 1
0

3
1 2
1 1
0

4
2 2 3
2 1 4
1 1
1 2

6
1 4
2 4 5
2 4 6
5 1 2 3 5 6
2 2 4
2 3 4

5
3 2 4 5
2 1 4
1 4
3 1 2 3
1 1

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

Sample Output 2

```
Case #1:
1 3 5 6
Case #2:
1 2
Case #3:
1 2 3 5
Case #4:
1 3
Case #5:
1 4
Case #6:
1 2 3 5 6
Case #7:
1 3
Case #8:
1 3 5 6
```

Problem D
Woodchucking

You may remember Lea's friend Nick, the biologist. Recently, he decided to do a study at a saw mill, *Jack Lumber Inc.*, a business producing lumber. He decided to do his study on the possible productivity increase of local industries by using rodents (woodchucks, mostly). So for a few weeks, Nick got in touch with his inner lumberjack and hacked away at everything that even remotely resembled a tree (the lumberjacks quickly learned not to come too close to Nick when he was armed with an axe). Once a month, the lumberjacks had to heave all the trees onto large conveyor belts at the saw mill, where they were cut up into small pieces by huge disk saws, ready to be shipped to whoever wanted to build barstools, wooden swords, toothpicks or whatever else. There were a lot of disk saws and conveyor belts and every tree took sawing time according to its height. This meant nobody knew on which conveyor the next tree had to be put to minimize the total sawing time. So these days were mostly ending in chaos and the buzzing and sawing took quite a while.

As a scientist, Nick could not resist to try to help them and quickly devised a computer program that computed the optimal distribution of trees to the saws. It took ages to compute the value, but now the lumberjacks could chuck down the trees, then sit around for a day and then Nick guided them to cut the trees up much faster.

Unfortunately, Handsome Jack, the boss at *Jack Lumber Inc.*, really did not like the fact that his employees sat around, not chucking wood, waiting for Nick's program to spurt out some numbers. Nick tried to reason with him, but it was no use. They had to go back to the old system until Nick came up with a better idea.

Can you help him devise a faster computation that cuts up the trees in at most 50% more time than his previous solution?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line. Each test case consists of a line containting two integers n and m. n is the amount of trees the lumberjacks felled during the course of the month, m is the amount of disk saws available. n lines follow, each containing a single integer l_i, the time it takes to cut up tree i.

Output

For each test case, output one line containing "Case #i: s" where i is its number, starting at 1, and s is at most 50% more than the minimal amount of time it takes to cut up every tree. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq m \leq 1000$
- $1 \leq l_i \leq 10^5$

Sample Input 1

```
2
3 2
2
8
4

4 3
1
1
2
4
```

Sample Output 1

```
Case #1: 8
Case #2: 4
```

Sample Input 2

```
5
4 5
1
3
7
4

4 2
4
9
5
5

3 2
10
9
9

4 4
6
4
8
7

3 4
8
4
10
```

Sample Output 2

```
Case #1: 7
Case #2: 13
Case #3: 18
Case #4: 8
Case #5: 10
```

Problem E
Dolphins

Recently Lea read a book about, among other things, the creation of the earth. In this scientifically accurate work it was detailed that the earth was a construction ordered by mice. Even more, it was said that dolphins are the second most intelligent species on earth, after the mice. This left Lea baffled: could it really be that humans are a mere third on the intelligence scale? Only one way to find out!

Lea wants to make sure that humans are superior to dolphins by comparing their DNA to the DNA of mice. Should she find out that they are closely related, it is clear that humans must be more intelligent than dolphins.

Comparing DNA is done by scoring: Lea has some DNA sequences from humans and some from mice. A sequence is a string consisting only of the letters A,C,T and G. Two equal length sequences are scored with a 4×4 scoring matrix whose rows and colums are labelled with A,C,T and G. An entry in row A and column T is then called $score(A, T)$. The sequences are scored by computing the score of the first letters of each sequence, the second letters, and so on, and summing up the result. For example, the sequences ATA and CGT would be scored as $score(A, C) + score(T, G) + score(A, T)$.

The matrix must have the following properties:

- All entries must be integers between -10 and 10, inclusively.
- It must be symmetric ($score(x, y) = score(y, x)$ for all $x, y \in \{A, C, T, G\}$).
- Diagonal entries must be non-negative ($score(x, x) \geq 0$ for all $x \in \{A, C, T, G\}$).
- The sum of all 16 entries must be 0.

Of course Lea wants to choose the matrix such that the sum of the scores she gets when comparing every human DNA sequence to every mouse DNA sequence is maximal. Can you compute the maximal score that she can achieve?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

The first line of each test case contains two integers n m, the number of human and mouse DNA sequences. The next n lines contain one human DNA sequence each. The last m lines contain a mouse DNA sequence each.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the maximal score that can be reached. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq m, n \leq 200$
- A sequence consists of at most 100 letters of A,C,T and G.
- For each test case, all sequences have equal length.

Sample Input 1

```
2
1 1
ACAC
ATGC

2 2
AAA
CTG
TCC
GGT
```

Sample Output 1

```
Case #1: 40
Case #2: 0
```

Sample Input 2

```
5
3 3
ACACACTGGTCTATACTTCG
ACTCGCTGAAGTTAATTACC
ACTGTGCGTCCAAGGGTAAT
ATCGGACAGATCACGCCCCT
ATGGTGATATTCAATGCTGT
CATATCGTACTAGGGTAAAG

2 5
CTATTTAGTT
AAACTGTAAT
GATTGATTTG
CTAGAGACCA
ATTGCTCCCG
GAGTACTAAA
TGCCGTTCTT

3 2
GGACGCG
ATTCGAT
ATAGCGA
TCACCCG
CCCACAG

4 5
TACG
GGAG
ACCC
GTCA
GCAC
GCGG
GAAG
GGTA
ATCT

3 2
CGTATGTCCCGCGA
GCAAGTGCGCTTTG
CAGGTTGGGGTCAA
GGTAAGGGGAAAAC
ATAAGCTTGCTCTC
```

Sample Output 2

```
Case #1: 330
Case #2: 370
Case #3: 150
Case #4: 240
Case #5: 150
```