hash code

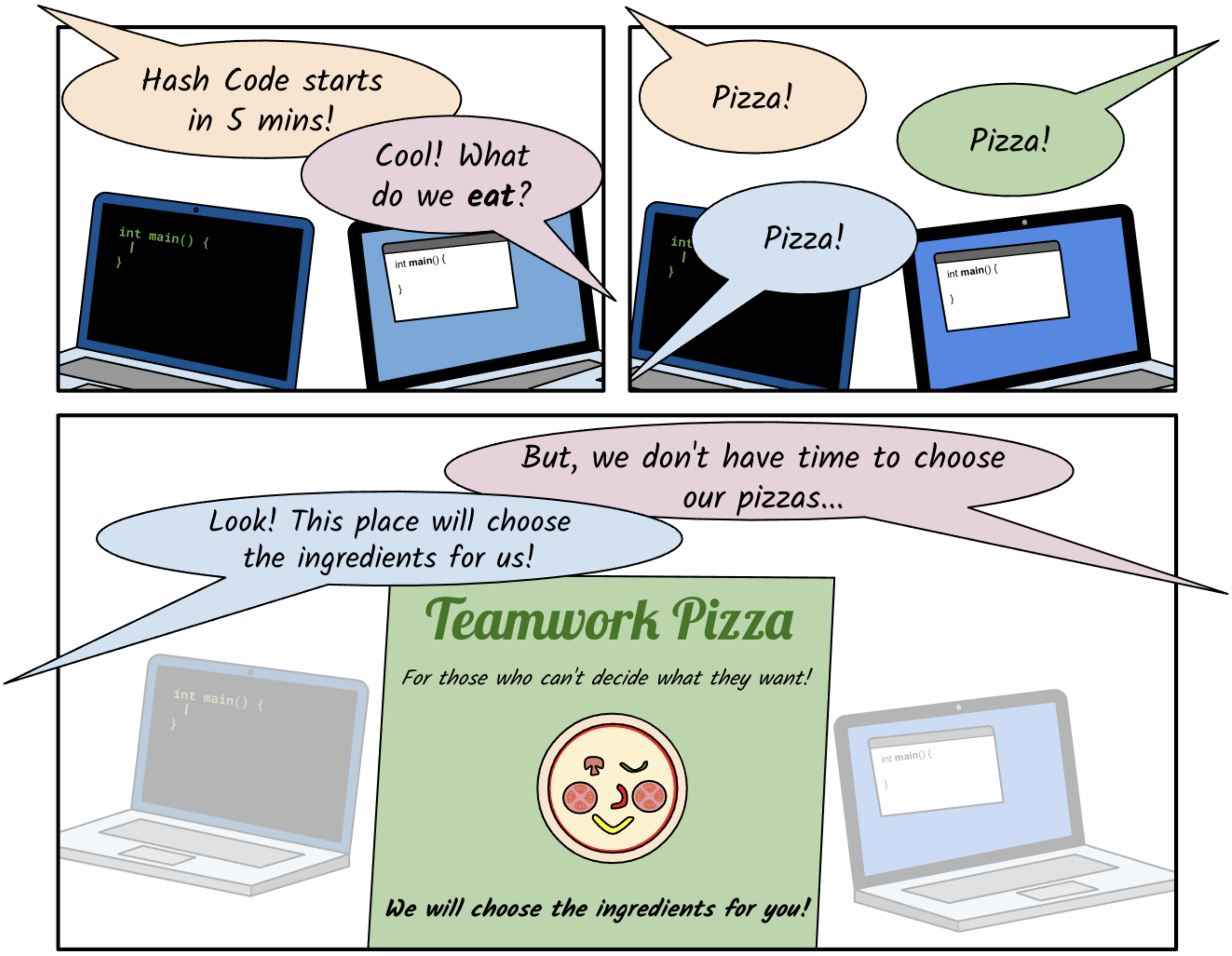
Even more pizza

Hash Code practice problem

Introduction

Isn't it fun to share pizza with friends? But, sometimes you just don't have enough time to choose what pizza to order. Wouldn't it be nice if someone else chose for you?

In an **imaginary** world...


Hash Code starts in 5 mins!
Cool! What do we **eat**?
Pizza!
Pizza!
Pizza!
But, we don't have time to choose our pizzas...
Look! This place will choose the ingredients for us!
Teamwork Pizza
For those who can't decide what they want!
We will choose the ingredients for you!

Problem description

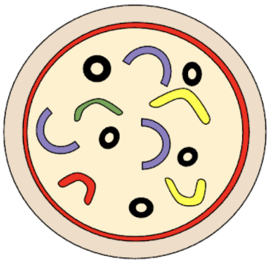
Task

Help the imaginary pizzeria choose the pizzas to deliver to Hash Code teams. And since we want everyone to enjoy their food, let's try to deliver to each team, as many different ingredients as we can.

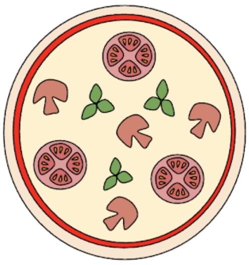
Pizza

Expecting many hungry customers, the pizzeria has already prepared some pizzas with different ingredients. Each pizza can be delivered to at most one team. There can be multiple pizzas with the exact same set of ingredients.

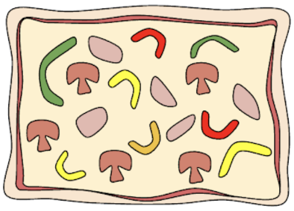
For **example**, there are 5 pizzas available in the pizzeria:



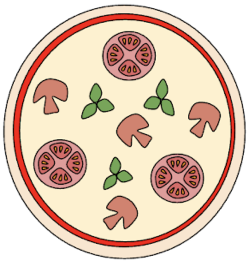
Pizza 0



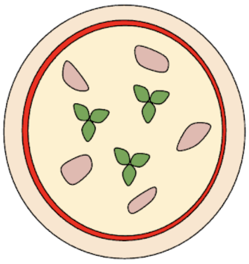
Pizza 1



Pizza 2



Pizza 3



Pizza 4

Pizza 0: onion, pepper, olive
Pizza 1: mushroom, tomato, basil
Pizza 2: chicken, mushroom, pepper
Pizza 3: tomato, mushroom, basil
Pizza 4: chicken, basil

Note that Pizzas 1 and 3 have the same ingredients, even though they are mentioned in different order.

Teams

Teams of 2, 3, or 4 people all ordered pizzas. Each team ordered one pizza per team member, but did not specify what ingredients to put on the pizzas. The pizzeria might not deliver to a team (no pizzas are sent to that team). However, if the order is delivered, exactly one pizza should be available per person. For example, it is an error to send 3 pizzas to a 4-person team.

Goal

Given the description of the pizzas available, and the number of teams of 2, 3, or 4 people that have ordered, decide which pizzas to send to each of the teams. The goal is to maximize, per team, the number of different ingredients used in all their pizzas.

For **example**, if we deliver to a 3-person team Pizzas 0, 2 and 3, there will be 7 different ingredients (9 ingredients in total, but **pepper** and **mushroom** occur twice):

- Pizza 0
  - **onion**
  - **pepper**
  - **olive**
- Pizza 2:
  - **chicken**
  - **mushroom**
  - ~~pepper~~ (is already on Pizza 0)
- Pizza 3:

- **tomato**
- ~~mushroom~~ (is already on Pizza 2)
- **basil**

Input data set

The input data is provided as a data set file - a plain text file containing exclusively ASCII characters with lines terminated with a single ‘\n’ character (UNIX-style line endings).

File format

The first line of the input file contains the following integer numbers separated by single spaces:

- **M** $(1 \le M \le 100\,000)$ - the number of pizzas available in the pizzeria
- **T$_2$** $(0 \le T_2 \le 50\,000)$ - the number of 2-person teams
- **T$_3$** $(0 \le T_3 \le 50\,000)$ - the number of 3-person teams
- **T$_4$** $(0 \le T_4 \le 50\,000)$ - the number of 4-person teams

The next **M** lines describe the pizzas available. Each line contains (space separated):

- an integer **I** $(1 \le I \le 10\,000)$ - the number of ingredients,
- followed by the list of **I** ingredients - Each ingredient consists of lowercase ASCII letters and dash (-) characters, and its length can be between 1 and 20 characters in total. Each ingredient in a pizza is different, but the same ingredient can appear on different pizzas.

Example

Input file	Description
`5 1 2 1`	5 pizzas, 1 team of two, 2 teams of three, and 1 team of four
`3 onion pepper olive`	Pizza 0 has the given 3 ingredients
`3 mushroom tomato basil`	Pizza 1 has the given 3 ingredients
`3 chicken mushroom pepper`	Pizza 2 has the given 3 ingredients
`3 tomato mushroom basil`	Pizza 3 has the given 3 ingredients
`2 chicken basil`	Pizza 2 has the given 2 ingredients

Submissions

File format

The first line of the submission file contains a number **D** $(1 \le D \le T_2 + T_3 + T_4)$, representing the number of pizza deliveries.

The following D lines contain descriptions of each delivery. Each line contains the following integer numbers separated by single spaces:

- **L** $(2 \le L \le 4)$ - the number of people in the team
- followed by the list of pizzas, P_1 ... P_L - the space separated indexes of the pizzas delivered to that team

Even though it's nice to deliver pizzas to all teams, it is allowed to make fewer deliveries than the number of teams. However, making more deliveries than the number of teams is an error. It is also an error to make more deliveries to 2, 3 or 4-person teams than the corresponding number of teams provided in the input file: the number of lines with L=N, should not be greater than T_N.

Example

`Submission file`	`Description`

`2`
`2 1 4`
`3 0 2 3`
	Pizzas are delivered to 2 teams
	A 2-person team will receive Pizza 1 and Pizza 4
	A 3-person team will receive Pizza 0, Pizza 2 and Pizza 3

Validation

In order for the submission to be accepted:

- each pizza must be part of at most one order,
- for all N-person teams, either nobody or everybody receives a pizza,
- there are T_N or less deliveries to teams of N people.

Scoring

For each delivery, the delivery score is the **square** of the total number of different ingredients of all the pizzas in the delivery. The total score is the sum of the scores for all deliveries.

For **example**, with the example input file and the example submission file above, there are

- 4 ingredients delivered to the two-person team (mushroom, tomato, basil, chicken). The score for that team is 4^2 = **16**
- 7 ingredients delivered to the tree-person team. The score for that team is 7^2 = **49**.
- (The score is **0** for the two teams that didn't have their order delivered)

The total score is 16 + 49 = **65**.

Note that there are multiple data sets representing separate instances of the problem. The final score for your team will be the sum of your best scores for the individual data sets.