

## Huawei Challenge in Innopolis

22-24 October 2021 | Innopolis University

### Rule-match problem

#### Problem statement

Rule-match problem is a classic problem in the Firewall filtering. A rule, as well as a key, has several dimensions. A key matches a rule if every dimension of the key matches the corresponding dimension of the rule. If the key matches several rules, you should return first rule in the list. A dimension of the rule can be:

- *Ternary*: The value of this dimension is a list of 0, 1, \*-bits, for the rule, and only 0, 1-bits, for the key. A key matches the rule in this dimension if every 0-bit of the key corresponds to 0 or \*-bit of the rule and every 1-bit of the key corresponds to 1 or \*-bit. For example, 0011 matches 0 \* 1\* and 00 \*\*.
- *Range*: The value of this dimension is a range in decimal system. A key matches the rule in this dimension if the key is inside the range including borders. For example, key 15 matches 11 – 20 rule.

#### Input

One input.txt file (unsorted rules and keys). First line is a number of rules, then each line is a rule. After all rules there is a line with a number of keys and then lines with keys are going. Dimensions are separated by space. Number of rules is no more than 150K, number of keys is no more than 15K.

#### Output

One output.txt file where each line corresponds to the matched rule's index (starting from 0). If there is no match, write -1. Number of lines in the output file equals number of keys.

#### Example

Ternary dimensions restricted to 5-bit instead of 32-bit.

input.txt
4
11*** **00* 11-21 112-122
***** 00*10 10-11 11-25
00*11 *1*0* 1000-1200 0-21
***** 111** 0-45 98-345
4
00111 01100 1000 12
10101 11111 21 99
11001 10001 13 123
01010 00010 11 25

output.txt
2
3
-1
1

## Huawei Challenge in Innopolis

22-24 October 2021 | Innopolis University

### Restrictions

- Language is C/C++ (-std=c++11)
- One thread, no parallelization.
- Only standard library could be used.
- Number of dimensions is no more than 5 and is equal along all of the data (including key data). Type of each dimension (ternary or range) is also fixed along all of the data.
- Ternary keys are 32-bit (ternary bit) long, range bounds are non-negative integers 0-65535.
- Memory limitation 500,000 KB.
- Time limitation 200,000 ms

### Compilation command

```
g++ -O2 -Wl,--stack=536870912  
-D__USE_MINGW_ANSI_STDIO=0 -x c++ -std=c++11  
-static <source>.cpp
```

### Description of the algorithm

Each team, in addition to the algorithm itself, is required to provide a description of the idea and concept embedded in the developed algorithm. The description is provided in .docx format in English. The description should not exceed 5 pages.

### Scoring

Two data samples are prepared for the competition. The first input file, as well as its correct output file, are provided to the teams. On the same pair of files, the PCMS platform performs a preliminary assessment of each of the components of the score for teams' submissions. After the end of the time allotted for the development of the algorithm, the final result is summed up on the basis of the last teams' submission and the second data sample.

In other words, the current score of the submission observed by the teams on the PCMS platform is preliminary, since it's carried out on the basis of the first sample. The final score may differ greatly from the preliminary one due to a qualitatively different distribution in the input data. Teams should carefully work out their algorithm and not get hung up on specific data from the first sample.

$$\text{Final score} = 1000000 \times (10 \times \text{error} + 5 \times \text{speed} + 2 \times \text{memory})$$

## Huawei Challenge in Innopolis

22-24 October 2021 | Innopolis University

---

- **error** = accuracy of the algorithm counted as the number of correct output lines divided by number of keys (float number in [0,1] with an accuracy of 6 decimal places). If the accuracy of the algorithm from the team is less than 0.95 then the measurement stops and the score of 0 is set. The minimum value for 10 runs of the algorithm is taken into account.
- **speed** = normalized speed of the algorithm (980 ms divided by the speed (in ms) of the algorithm from the team with an accuracy of 6 decimal places). If the speed of the algorithm from the team exceeds 200,000 ms then the measurement stops and the score of 0 is set. The maximum value for 10 runs of the algorithm is taken into account.
- **memory** = normalized memory usage of the algorithm (13000 KB divided by the memory (in KB) of the algorithm from the team with an accuracy of 6 decimal places). If the memory of the algorithm from the team exceeds 500,000 KB then the measurement stops and the score of 0 is set. The maximum value for 10 runs of the algorithm is taken into account.