

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 2 (Линейный поиск)

Н. Наилучший запрет

Ограничение времени	3 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Константин и Михаил играют в настольную игру «Ярость Эльфов». В игре есть n рас и m классов персонажей. Каждый персонаж характеризуется своими расой и классом. Для каждой расы и каждого класса существует ровно один персонаж такой расы и такого класса. Сила персонажа i -й расы и j -го класса равна a_{ij} , и обоим игрокам это прекрасно известно.

Сейчас Константин будет выбирать себе персонажа. Перед этим Михаил может запретить одну расу и один класс, чтобы Константин не мог выбирать персонажей, у которых такая раса **или** такой класс. Конечно же, Михаил старается, чтобы Константину достался как можно более слабый персонаж, а Константин, напротив, выбирает персонажа посильнее. Какие расу и класс следует запретить Михаилу?

Формат ввода

Первая строка содержит два целых числа n и m ($2 \leq n, m \leq 1000$) через пробел — количество рас и классов в игре «Ярость Эльфов», соответственно.

В следующих n строках содержится по m целых чисел через пробел. j -е число i -й из этих строк — это a_{ij} ($1 \leq a_{ij} \leq 10^9$).

Формат вывода

В единственной строке выведите два целых числа через пробел — номер расы и номер класса, которые следует запретить Михаилу. Расы и классы нумеруются с единицы. Если есть несколько возможных ответов, выведите любой из них.

Пример 1

Ввод

Вывод

2 2
1 2
3 4

2 2

Пример 2

Ввод

Вывод

3 4
1 3 5 7
9 11 2 4
6 8 10 12

3 2

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

```
1 # считываем данные
2 n, m = map(int, input().split()) # количество рас и классов
3 max_in_row = [0] * n * 2
4 character_grid = [[]] * n
5
6 for i in range(n):
7     cur_race = list(map(int, input().split()))
8     character_grid[i] = cur_race
9     max_1, idx_max_1, max_2, idx_max_2 = -1, -1, -1, -1
10    for j in range(m):
11        if max_1 <= cur_race[j]:
12            max_2 = max_1
13            idx_max_2 = idx_max_1
14            max_1 = cur_race[j]
15            idx_max_1 = j
16        elif max_2 < cur_race[j]:
17            max_2 = cur_race[j]
18            idx_max_2 = j
19
20    max_in_row[i*2] = (max_1, i, idx_max_1)
21    max_in_row[i*2+1] = (max_2, i, idx_max_2)
22
23 max_in_row.sort(key=lambda x: -x[0])
24
25 answer_row, answer_col = -1, -1
26 if max_in_row[0][1] != max_in_row[1][1] and max_in_row[0][2] != max_in_row[1][2]:
27     if max_in_row[2][1] == max_in_row[0][1]:
28         answer_row, answer_col = max_in_row[0][1], max_in_row[1][2]
29     elif max_in_row[2][2] == max_in_row[0][2]:
30         answer_row, answer_col = max_in_row[1][1], max_in_row[0][2]
31     elif max_in_row[2][1] == max_in_row[1][1]:
32         answer_row, answer_col = max_in_row[1][1], max_in_row[0][2]
33     elif max_in_row[2][2] == max_in_row[1][2]:
34         answer_row, answer_col = max_in_row[0][1], max_in_row[1][2]
35     else:
36         answer_row, answer_col = max_in_row[0][1], max_in_row[1][2]
37
38 elif max_in_row[0][1] == max_in_row[1][1]:
```

Отправить

Предыдущая

Следующая