ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger
Prof. Dr. Emo Welzl
Prof. Dr. Peter Widmayer

Algorithms Lab

Exercise – *Punch*

Paul's birthday is just around the corner and his friends decide to throw him a surprise party. Since the most important thing for a good party is drinks, they decide to prepare a delicious punch as a mixture of several beverages. Hence, they go to a store which sells n distinct beverages (the store has unlimited supply for each beverage). Each beverage is sold at a certain price (e.g. 2 SFr.) in a certain volume (e.g. 1 l). There are k people coming to the party and there should be at least one liter of punch for each of them. Ufortunately, Paul's friends are broke so they want to minimize the cost of the punch. However, if there are different ways to make the cheapest punch, then they want to maximize the number of distinct beverages (for the taste).

Your task is to compute the cost and the number of distinct beverages.

Input The first line of the input contains the number `t` of test cases with $1 \leq t \leq 200$. Each test case is described as follows:

- It starts with a line that contains two integers `n` and `k`, where n is the number of distinct beverages in the store ($1 \leq n \leq 100$) and k is the number of people coming to the party ($1 \leq k \leq 10^4$).
- The following n lines each describe a certain baverage. They contain two integers `c` and `v`, where c is the cost of the beverage ($1 \leq c \leq 10^4$) and v is the volume of the beverage in liter ($1 \leq v \leq 10^4$).

Output For each test case you should output, on a single line and separated by a space, the cost of a *cheapest* punch and the *maximum* number of *distinct* beverages in a cheapest punch.

Points There are two test sets, worth 100 points in total.

1. For the first test set, worth 50 points, you may assume $k \leq 100$ and $v \leq 100$ for all beverages.
2. For the second test set, worth 50 points, there are no additional assumptions.

Corresponding sample test sets are contained in `test`i`.in/out`, for $i \in \{1, 2\}$.

Hint: Put `std::ios_base::sync_with_stdio(false)` as the first line of the main procedure for faster I/O.

Sample Input

```
2
2 4
1 1
1 2
3 4
1 1
2 2
2 1
```

Sample Output

```
2 1
4 2
```