min-cost max flow is not correct, it is possible that max gain is achieved at lower flow.

I think you can do it by adding some virtual node that have 0 cost, this will make the optimal solution have maximal flow.

This adding, intuitively, requires O(b*s)

But for a bipartite graph, how can we guarantee the max flow to be exactly min(b,s) by adding limited virtual edges?

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithms Lab HS22
Department of Computer Science
Prof. Dr. A. Steger, Prof. Dr. E. Welzl
cadmo.ethz.ch/education/lectures/HS22/algolab

my solution is that we can bypass b and add b to sink or source to s

Exercise – *Fleetrace*

AlgoSail is one of the largest competitions in single-handed sailing (meaning that there is only one sailor per boat) around the Globe. For this year's edition, Madame Vendée has been appointed to the role of a race director. Her plan is nothing else but to have the most spectacular fleet race ever. In order to achieve these lofty goals, she needs to effectively *pair* the *boat owners* with the *sailors*.

The boat owners announced a subset of sailors that they are willing to let sail on their boats. Conversely, the sailors announced on which boats they are willing to sail. From these statements, Mme Vendée has compiled a list of compatible *boat/sailor pairs*.

In the past few years, AlgoSail has become slightly boring. Maybe this is due to the fact that the previous race directors have always tried to have as many participants as possible. Thus in recent editions, the boats and sailors have been paired quite similarly, and the outcome of the race became very predictable.

This year, things are going to change: Having the goal of a superbly spectacular fleet race in mind, Mme Vendée has computed for each entry in her list of boat/sailor pairings a *spectacle coefficient*, based on a collection of past performances of the sailors in other races.

Help Mme Vendée to organize a fleetrace as spectacular as possible:
Your goal is not necessarily to have as many participating boat/sailor pairs as possible, but to maximize the sum of the spectacle coefficients of all participating pairs.

Input The first line of the input contains the number $t \leqslant 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that contains three integers b s p, separated by a space. They denote
  - b, the number of boat owners ($1 \leqslant b \leqslant 500$),
  - s, the number of sailors ($1 \leqslant s \leqslant 500$), and
  - p, the number of boat/sailor pairs on Mme Vendée's list ($1 \leqslant p \leqslant b \cdot s$ and $p \leqslant 5'000$).
- The following p lines describe the boat/sailor pairs and their spectacle coefficients. The i-th of these lines contains three integers b_i s_i c_i, separated by a space, denoting that sailor s_i could sail on boat b_i, with a spectacle coefficient of c_i (where $0 \leqslant b_i < b$, $0 \leqslant s_i < s$, $1 \leqslant c_i \leqslant 50$).
  No boat/sailor pair appears more than once, that is, $\forall\, i, j\colon\ i \neq j \Rightarrow (b_i, s_i) \neq (b_j, s_j)$.

Output For each test case, output a line with one integer C, the maximum sum of spectacle coefficients of any feasible boat/sailor pair selection. (A *selection* is a subset of Mme Vendée's list of pairs. It is *feasible*, if no boat and no sailor appears more than once in the selection.)

Points There are four groups of test sets, each of which is worth 25 points.

1. For the first group of test sets, you may assume that $b, s \leqslant 30$ and that every boat owner is willing to work with every sailor and vice versa (that is, $p = b \cdot s$).
2. For the second group of test sets, you may assume that $b, s \leqslant 30$.
3. For the third group of test sets, you may assume that $b, s \leqslant 100$ and that $p \leqslant 1'000$.
4. For the forth group of test sets, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3, 4\}$.

Sample Input

```
5
2 3 6
0 0 6
0 1 3
0 2 4
1 0 9
1 1 2
1 2 8
4 3 7
0 0 3
0 1 3
1 0 3
1 1 3
2 0 3
2 1 3
2 2 3
4 5 8
0 0 5
0 1 5
1 0 5
2 0 5
2 1 5
3 1 5
3 2 5
3 3 5
4 3 7
0 0 4
0 1 5
1 0 1
1 1 2
2 0 6
2 1 10
2 2 7
4 5 8
0 0 4
0 1 5
1 0 4
2 0 6
2 1 4
3 1 10
3 2 5
3 3 6
```

Sample Output

```
14
9
15
14
17
```