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

Exercise – *World Cup*

After more than 60 years of waiting, the most prestigious event in football, the so called *World Cup*, is to be held in Switzerland again. After a long and strenuous (and expensive!) campaign, targeted at several high-ranking IFAF[1] operatives, your company—the BEER® brewery—has secured exclusive selling rights for beer in all stadiums during the event. Being the good business person that you are, you try to get maximum profit out of this unique opportunity.

There are *warehouses* spread all over Switzerland. Each warehouse stores exactly one specific brand of BEER and it has a limited *supply* (in liters) of that specific brand. The different brands of BEER are characterized by their varying alcohol contents. Moreover, there are *stadiums*, in each of which exactly one match is played. Each stadium has a certain *demand* (in liters) of BEER that is consumed by the football fans who attend the corresponding match.

Unsurprisingly, IFAF is in great fear of riots if the football fans find themselves unable to quench their thirst. Part of your contract thus states that you must deliver the exact demanded quantity of BEER to each stadium, no more and no less. How you meet this obligation is up to you; in particular, you may sell any number of distinct brands of BEER at each stadium. There is one final catch, however: Again because of a fear of riots, your contract sets an *upper limit* (in liters) on the total amount of pure alcohol that is consumed at each stadium.

For every pair (w, s) of a warehouse w and a stadium s, you may transport $a_{w,s}$ liters of the corresponding brand of BEER from w to s, for any real number $a_{w,s} \geqslant 0$. Of course, the total amount of BEER transported out of w cannot exceed the supply of w. For each liter of BEER that is transported from w to s you generate a certain *revenue* $r_{w,s}$ in CHF. The numbers $r_{w,s}$ have been computed in advance, and you know that some of them might be negative.

Unfortunately, in the calculation of the numbers $r_{w,s}$ the mountainous terrain of Switzerland was not taken into account. This terrain is described by a set of *contour lines*[2]. For simplicity we assume that each contour line is a perfect circle. You estimate a loss of 0.01 CHF per liter of BEER that is transported across such a contour line, and you assume that going downhill or uphill incurs the same loss. Thus, the actual *profit* for transporting 1 liter of BEER from w to s is $r_{w,s} - (t_{w,s}/100)$, where $r_{w,s}$ is provided as above and $t_{w,s}$ is the number of contour lines that your delivery trucks traverse when driving from w to s. You know that every pair (w, s) has the property that the road between w and s traverses each contour line at most once.

Given all the constraints above, maximize the *total profit*, which is simply the sum of all profits made for each pair (w, s).

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 `n m c`, separated by a space. They denote, respectively, the number n of warehouses ($1 \leqslant n \leqslant 200$), the number m of stadiums ($1 \leqslant m \leqslant 20$), and the number c of contour lines ($0 \leqslant c \leqslant 10^6$).

[1] International Federation of Association Football.

[2] These are the lines that you typically see on a topographic map, to identify points with the same elevation.

The including relation ship of contour lines form a multi-tree, and the distance on this tree is the total contour line you have to cross.
or you can also denote each stadium or warehouse by a c-long vector, that denotes whether inside or outside, the manhaton distance is the number of warehouse, or equivalently, sum of xor.

each vector O((n+m)c) to get contour is O(cnm)

if we know the distance from root, then getting t is like

- The following n lines describe the warehouses $w_1, \dots, w_n$. Each line contains four integers `x y s a`, separated by a space and such that $|x|, |y| < 2^{24}$, $0 \leqslant s \leqslant 10^5$ and $0 \leqslant a \leqslant 100$. Here, (x, y) denotes the position of the corresponding warehouse, s denotes its supply, and a denotes the alcohol content (in %) of the brand of BEER stored there.
- The following m lines describe the stadiums $s_1, \dots, s_m$. Each line contains four integers `x y d u`, separated by a space and such that $|x|, |y| < 2^{24}$, $0 \leqslant d \leqslant 10^5$ and $0 \leqslant u \leqslant 10^5$. Here, (x, y) denotes the position of the corresponding stadium, d denotes its demand, and u denotes the upper limit on pure alcohol to be consumed there.
- The following n lines describe the estimated revenues $r_{w,s}$. Each line contains m integers $r_{w,1}$ ... $r_{w,m}$, separated by a space and such that $-10 \leqslant r_{w,s} \leqslant 10$. That is, the s-th entry in the w-th line denotes the number $r_{w,s}$.
- The following c lines describe the contour lines $\ell_1, \dots, \ell_c$. Each line contains three integers `x y r`, separated by a space and such that $|x|, |y| < 2^{24}$ and $0 < r < 2^{24}$. Here, (x, y) denotes the center of the corresponding contour line and r denotes its radius.

You may assume that the contour lines are pairwise disjoint, and that no warehouse or stadium is located directly on a contour line.

Output For each test case the corresponding output appears on a separate line. If it is possible to comply with your contract, the output is the maximum achievable total profit in CHF, rounded down to the next integer (the rounding is towards $-\infty$; for instance, -4.5 is rounded to -5). Otherwise, the output is `RIOT!`.

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 there are no contour lines ($c = 0$) and that all estimated revenues are zero ($r_{w,s} = 0$).
2. For the second group of test sets, you may assume that there are no contour lines ($c = 0$).
3. For the third group of test sets, you may assume that there is only a small number of contour lines ($c \leqslant 100$).
4. For the fourth group of test sets, you may assume that there are at most 100 contour lines that contain at least one warehouse or stadium in their interior.

so this contour lines are very sparse. how to avoid all these unnecessary computation?

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

Sample Input

```
5
1 1 0
0 0 20 5
3 0 20 1
1
1 1 0
0 0 19 5
3 0 20 1
1
```

```
1 1 0
0 0 20 6
3 0 20 1
1
1 1 2
0 0 20 5
3 0 20 1
1
0 0 1
0 0 2
```

```
2 2 0
0 0 20 5
0 3 20 10
3 0 20 2
3 3 20 1
1 -1
-1 1
```

Sample Output

```
20
RIOT!
RIOT!
19
-40
```

we can compute the delaunay of all ware house and stadium. If the closest is not in, then we discard this point. Then we construct a 100-vector, 100*20*200=400,000, 100*100=10,000. I think using a tree structure is possible, but is complex, and I skip it.

but how about the linear programming part? there are 20*200 = 4000 variable. 20x2(exact constrain) + 20 upperbound constrain.