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 – *Ceryneian Hind*

“πλὴν ἄνευ βίας καὶ κινδύνων διὰ τῆς κατὰ τὴν ψυχὴν ἀγχινοίας τὸν ἄθλον τοῦον κατειργάσατο.”

One thing is certain, that he accomplished this Labor by his sagacity of mind, without the use of force and without running any perils.

The Library of History *Book II by* Diodorus Siculus

After Hercules successfully slayed the Nemean Lion and the Lernaean Hydra, Eurystheus finally realized that our hero was too good at fighting monsters. Therefore, Eurystheaus decided on a more difficult task for the third labor: capturing the Ceryneian Hind, a beast so fast that it could outpace a deadly arrow.

Hercules knew that despite being a demigod, he could not match the speed of the Hind. Therefore, he made a cunning plan: he would chase the Hind to a *semi-dead end* S, a district without any path leaving it. Then Hercules would run after the Hind in this district incessantly until the beast was exhausted and Hercules could capture it.

To choose where to chase the hind to, Hercules has a map of ancient Europe. This map can be regarded as a directed graph, whose vertices correspond to locations (forests, mountains, meadows, etc.) and whose edges correspond to paths that connect these locations. Furthermore, not all locations are ideal for capturing a sleeping beast, for instance, doing so in a swamp is more challenging than on an open field. To account for this, Hercules carefully scored each location on the map according to how convenient it is to conduct the capture there.

To maximize the chance of success, Hercules hence needs to find a semi-dead end S with a maximum convenience score. Formally, a semi-dead end is a non-empty subset S of vertices such that no edge leaves S. The convenience score of a semi dead-end S is the sum of the convenience scores of all the vertices in S. It may happen that no suitable semi-dead end exists: if there is no semi-dead end with a positive convenience score, then Hercules will give up the current plan and try to find another solution.

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 containing two integers `n m`, separated by a space. They denote
  - n, the number of locations on the map ($2 \leqslant n \leqslant 500$);
  - m, the number of directed paths on the map ($1 \leqslant m < 250,000$).
- The following line contains n integers, separated by a space. The i-th integer s_i denotes the convenience score of the i-th location ($|s_i| \leqslant 2^{10}$).
- The following m lines describe the directed paths. Each of the lines contains two integers `u v`, separated by a space. They denote
  - u, the starting location of the path ($0 \leqslant u \leqslant n-1$);

- v, the ending location of the path ($0 \leqslant v \leqslant n-1$ and $v \neq u$).

Output For each test case the corresponding output appears on a separate line. If the map does not contain any semi-dead end with a strictly positive convenience score, the output is "`impossible`". Otherwise, the output is the maximum convenience score over all semi-dead ends on the map.

Points There are four groups of test sets. For each group there is also a corresponding hidden test set, each worth 5 points. Overall, you can achieve 100 points.

1. For the first group of test sets, worth 25 points, you may assume that the map is a polytree[1].
2. For the second group of test sets, worth 25 points, you may assume the convenience score of any location is either -1 or 1, and all paths are oriented from a location with a positive score to a location with a negative score.
3. For the third group of test sets, worth 15 points, you may assume all locations have non-zero convenience scores, and that there are no paths between two locations with both positive scores nor between two locations with both negative scores.
4. For the fourth group of test sets, worth 15 points, there are no additional assumptions.

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

Sample Input

```
4
5 4
2 -3 -1 3 2
0 1
0 2
2 3
4 2
4 3
1 1 -1 -1
0 2
0 3
1 3
4 3
3 1 -1 -2
0 2
0 3
1 3
5 5
3 3 -2 -2 -1
0 2
1 4
2 3
3 2
3 4
```

Sample Output

```
4
impossible
1
2
```

if we use the idea in "asterix in switzerland", we want a subset that connect to all its outcoming node. then we have to make all edges have "infinity capacity" to ensure, after the maximum flow, all out coming edges have residue capacity.

but how to prove this is optimal??? and what is the net gain?

flow = capacity out = sum positive s_i out of S + sum negative s_i inside s = sum of all s_i positive - (sum all s_i inside S (with sign)) --> this is minimum

[1] A polytree is a directed acyclic graph whose underlying undirected graph is a tree. Namely, if we turn each edge of a polytree into an undirected one, the resulting graph is both connected and acyclic.