Algorithms and Data Structures

Greedy algorithms in Barbieland (`barbie`)

Problem text

Original slides on: judge.science.unitn.it/slides/asd23/prog1.pdf

Barbieland

In Barbieland everything flows smoothly and efficiently. Its inhabitants are all happy, and life is simple and devoid of counterexamples that invalidate solutions based on greedy techniques. In fact, every inhabitant is convinced that making the most inviting decision from time to time always leads to solving all their problems correctly. However, one day Barbie wakes up with a dilemma: *what if it wasn't true that greedy algorithms always lead to an optimal solution?* This question leads her towards an inner crisis that is not easy to manage.

Revelation

Barbie decides to look for an answer and relies on *Barbie Professor of Algorithms*, known for being the most nonconformist inhabitant of Barbieland when it comes to algorithms. Barbie Professor of Algorithms confesses to her that, in reality, the problems that can be solved with greedy techniques are only a small part, and there is a sea of problems out there that need more sophisticated solutions. After this revelation, Barbie Professor of Algorithms invites Barbie to practice, to master the art of problem solving. He then advises her to head to Algorithm, a place full of new problems to discover and not limited by the techniques she already knows.

Algoritmia

The World of Barbie is in fact made up of a series of cities. Each pair of cities may or may not be connected by a road with a certain travel time. From one city you can always reach all the others, potentially crossing one or more intermediate cities. Here Barbieland is indicated by B, Algoritmia with A and the rest of the cities with an identification number. Travel times are full positive. Given the journey to be faced, Barbie Professor of Algorithms lends Barbie her personal car: the *GraphCruise*. This car has a peculiarity: once you have selected the city of departure and the city of destination, it will always arrive following a route of minimum cost among all possible routes. In addition, he gives Barbie a book from which to study new algorithmic techniques.

An enemy

Everything looks perfect. However, the notorious enemy group of efficient algorithms, the *Miseri Tremendously Tedious and Slow Algorithms*, has received a tip and is ready to foil Barbie's plan. The M.A.T.T.E.L. has sent some henchmen around the various cities. If the GraphCruise passed through an occupied city, it would be stopped immediately. Obviously, the GraphCruise has its own rules, which it will follow perfectly: the car will choose one of the shortest routes considering the whole map. It could therefore decide autonomously that it should pass through an occupied city, in the event that there was a minimum cost route that provides for it. Given the autonomous driving of the GraphCruise, Barbie will spend the trip studying from the book received. Obviously, the more time you study, the more prepared you will be for arrival. Fortunately, Barbie Professor of Algorithms has a secret power: she is able to stretch all the roads in Barbie's World at will in one fell swoop, but all increases in road travel times will be of the same K *value*.

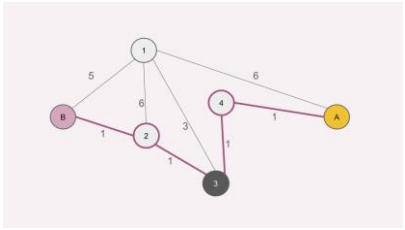


Figure 1: Example map: Here the minimum route is marked in pink, and would pass through an occupied city.

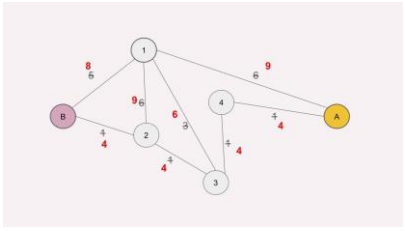


Figure 2: Example map: Barbie Professor of Algorithms has increased the travel time of each road by 3.

Question

Barbie Professor of Algorithms will try to use her power to help Barbie, allowing her to spend as much time as possible studying in the car. Help her in her work.

1. **What is the maximum value of K that Barbie Professor of Algorithms can choose so that the route chosen by the GraphCruise must necessarily pass through unoccupied cities?**
2. Also, **What is a minimum cost path when the Professor decides not to use her power?**

Examples

Example 1: All roads with cost 1

In the example shown in Figure 3, there are 11 cities and 15 roads, the streets all have travel time 1, the gray nodes are the occupied ones, Barbieland is the pink node, Algoritmia is the yellow node. The shortest route (cost 2) is the one marked in pink, it is unique and does not pass through gray nodes. Any value of K would not change the minimum path: K is unlimited (output -1).

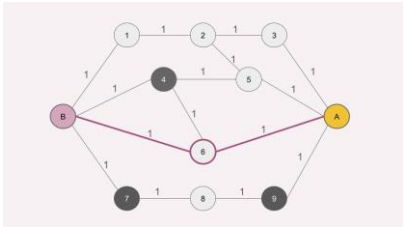


Figure 3: Case with all equal weights: the minimum path is the pink path and K is unlimited.

Example 2: K does not exist

Figure 4, Graph same as before, but node 6 is now occupied. There are no other minimum paths than B–6–A. Any value of K would not change the fact that B–6–A is the minimum path: output -2.

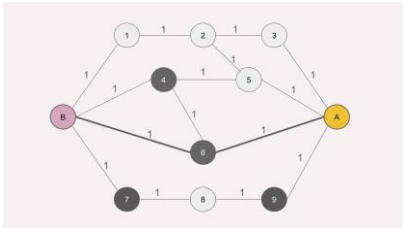


Figure 4: Case where K does not exist: the minimum path always passes through gray nodes.

Example 3: All free cities

In the case in Figure 5 there are no occupied cities. Of course, any value of K is fine, so K is unlimited (output -1). The minimum path when $K = 0$ is B–4–5–A of cost 6.

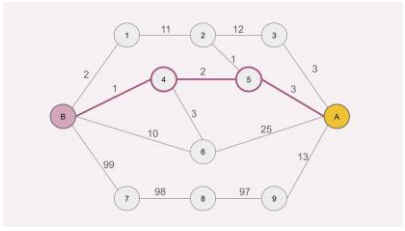
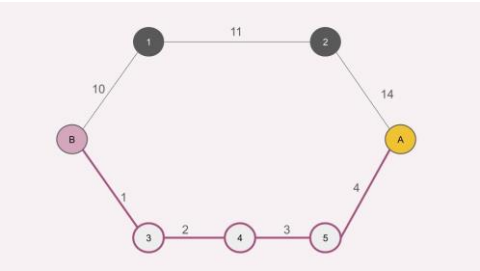


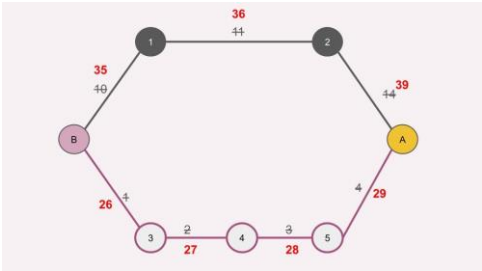
Figure 5: Case where there are no occupied cities: K is unlimited.

Example 4: Disjoint Paths

In the case in Figure 6a, the minimum path has a cost of 10 and involves passing only through free nodes. There is another route to get from B to A that costs 35, and involves passing through occupied nodes. We select $K = 25$ and update all the weights (red in Figure 6b). At this point the two routes are both of a minimum cost of 110. So Barbie would risk taking a minimal route that passes through an occupied city. $K = 24$ is the maximum value of K so it is guaranteed that Barbie passes only through unoccupied nodes.



(a) A minimum path in pink.



(b) Travel times increased with $K = 25$.

Figure 6: Example with disjoint paths.

Example 5: General case

In Figure 7 we propose a more general example. With $K = 0$ we have that the minimum path is unique and passes through unoccupied cities. The path is B–2–3–4–A and has a cost of 4. There are other routes to get from B to A, but not of minimal weight and possibly passing through busy cities.

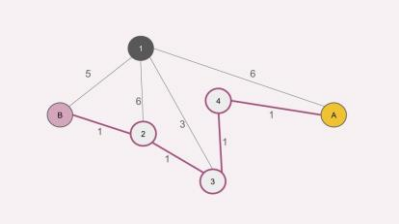
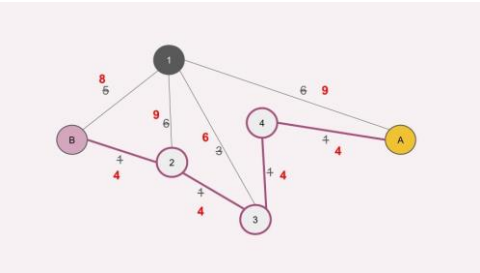
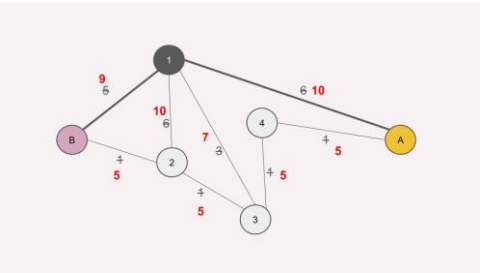


Figure 7: General case: the minimum path is the pink one.

Let's try to choose $K = 3$ and update the weights (red, Figure 8a). The minimum path remains B–2–3–4–A, but now has 16 costs. Let's try to choose $K = 4$ and update the weights (red, Figure 8b). At this point, the minimum path is B–1–A with cost 19. However, it should be noted that this route passes through an occupied city. It is therefore not a valid value of K. Path B–2–3–4–A now has a 20 cost.



(a) K=3.



(b) K=4

Figure 8: General example, two attempts for K.

Other examples of incorrect solutions

In Figure 9a we have an example where K is unlimited. You have to find the minimum path when $K = 0$. Let's say you output path B–1–A, highlighted in pink, with a cost of 11. However, there is the path highlighted in black, which costs 4. Your output is incorrect, as it is not a minimal path.
Looking at Figure 9b, suppose you have given B–2–A and K unlimited as the minimum path. The path is correct, one between B–2–A and B–1–A is fine, however finding K is impossible. For any value of K, both paths would maintain the same cost, ergo there would always be a minimum cost path with a city occupied. Figure 9c, example we have already seen. Suppose you have identified $K = 2$; with this value of K the minimum path does not pass through any occupied city. However, K is not maximal, in fact even for $K = 3$ the minimum cost path does not pass through occupied cities.

Input/Output

Assumptions and test cases

Recruitment

1. $2 \leq C \leq 1.000$
2. Barbieland is at node 0, Algoritmia is at node $C - 1$
3. Any other city is identified by a number between 1 and $C - 2$
4. $0 \leq M \leq C - 2$

- Barbieland e Algoritmia are always NOT occupied
- $0 \le S \le 10.000$

1. For the travel time wi of the i-th street is worth $1 \le wi \le 100.000$
2. Every graph is indirect.
3. Each graph is connected.
4. There are no self-loops, i.e. arcs (a,a) where a is a city
5. There are no double arcs, i.e. there is at most one triple (a,b,w) for a given unordered pair $\{a,b\}$ **Test cases** There are 20 test cases in total:
6. In 6 cases out of 20, all arches have the same travel time;
7. In 4 cases out of 20, either no city is occupied, or all cities are occupied (B and A excluded); • In 4 cases out of 20, the routes from B to A are all disjointed, i.e. they do not share intermediate cities;
8. In the remaining cases there are no particular limitations.

The time and memory limits are:

1. Maximum time limit: 1 second;
2. Maximum memory: 16 MB.

Scores and Checker

Each test case is worth 5 points if complete (compulsory and optional part), 3 points if partial (compulsory part only). The maximum score is 100 points.
In case of incorrect output, you always get 0 **points**: * if K is incorrect, you get 0 **points**. * if K is correct, but the printed path is not possible, you get 0 **points**. * if K is correct, but the printed path is not minimal, you get 0 **points**.

Evaluation

For project evaluation:

1. Counts the score of the **last source** sent to the system;
2. The project is passed with a score of not less than 30 points;
3. There is a limit of 50 subpositions per group;

Sample datasets

You can find an equivalent dataset to practice with locally: judge.science.unitn.it/slides/asd23/dataset_barbie.zip.
Note: The sample dataset outputs only the first row, which indicates the value of K, without any further information.

Esempi di input/output

File input.txt	File output.txt
6 0 1 5 0 2 1 1 2 6 1 3 3 1 5 6 2 3 1 3 4 1 4 5 1 0	-1 5 0 2 3 4 5

File input.txt	File output.txt
20 31 9 7 7 1 7 1 4 1 7 5 2 9 1 6 6 8 1 3 7 1 1 14 17 3 8 5 1 9 6 7 1 9 1 3 8 1 3 5 3 1 4 8 11 15 1 18 10 6 14 19 8 1 1 4 3 2 5 19 17 7 2 8 1 15 10 3 1 3 2 12 18 6 1 5 8 1 3 3 0 1 6 5 1 3 6 1 6 1 9 4 4 1 4 5 1 7 8 9 12	-2 5 0 1 6 6 1 1 9

File input.txt	File output.txt
7 0 1 1 0 3 1 1 2 1 2 6 1 3 4 2 4 5 3 5 6 4 2 1 2	24 5 0 3 4 5 6

File input.txt	File output.txt
7 10 1 0 3 2 4 3 2 1 3 5 0 1 4 6 6 2 6 8 5 1 1 4 0 8 3 4 5 5 3 9 2 3 4	-2 5 0 5 1 2 6