Prima

**Task:**

Timur and his friends, had arrived in the summer at their old cottages, decided to arrange a game for the duration of their vacation. They organized a team to secretly help the residents of the country town in their daily works. The holiday village is quite large, and the houses in which Timur's friends live are far from each other. How to quickly send messages to each other? How to collect guys for meeting?

Timur decided to lay a rope telegraph which connect all the cottages of Timur`s friend. Total number of houses is N. On the map, the guys calculated the coordinates of each house (Xi, Yi) in integers and wrote them out on paper. They took one meter per unit of coordinate measurement. However, the got a question: which houses should be connected with a rope telegraph so that the connection is between all the houses (possibly through other houses), and the total length of all the ropes is as small as possible?

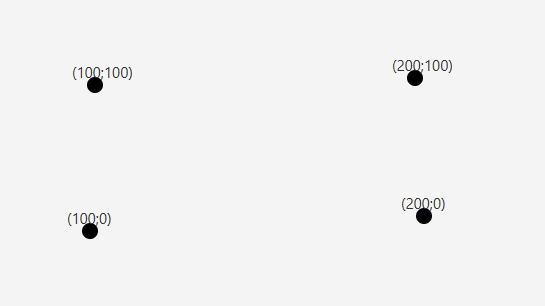
**Decision:**

Let Timur friends’ houses Will be nodes of the graph, the ropes between them - edges, and the lengths of the ropes as the weights of the edges. Now we are faced with the problem of a minimal spanning tree.

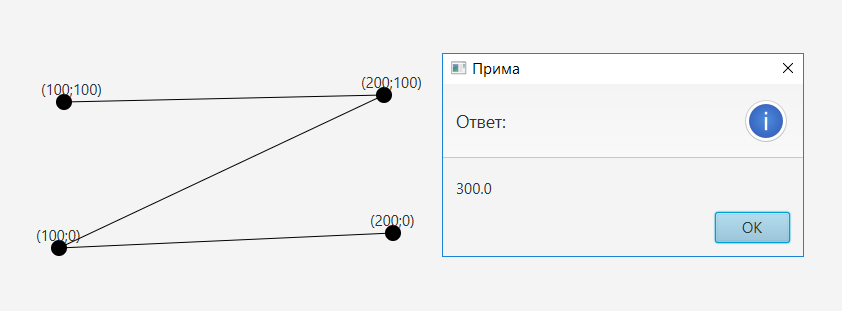
In this case, the initial graph is complete, that is, there is an edge between any two of its vertices, since according to the conditions of the problem, the rope can be stretched between any two houses. For the input data of the algorithm, we represent the graph in the form of an adjacency matrix.

**Implement:**

**Input data:**



Pic. 1 – Input data for Prima alg.

**Output data:**

Pic.2 – Output data of Prima alg.

Kruskal's algorithm

**Task:**

Andrey works as a system administrator and plans to create a new network communication between computers in his company. There will be N hubs in total, they will be connected to each other using cables.

Since every employee in the company must have access to the entire network, each hub must be reachable from any other hub - perhaps through several intermediate hubs. There are various types of cables and shorter cables are cheaper. It is necessary to make a network plan (connection of hubs) so that the maximum length of one cable is as small as possible. There is another problem - not every pair of hubs can be directly connected due to compatibility problems and the geometric constraints of the building. Andrey will provide you with all the necessary information about the possible connections of the hubs. It is necessary to help Andrey find a way to connect the hubs that will satisfy all the above conditions.

**Inputs:**

The input is an incomplete graph, with weights for each edge.

**Outputs:**

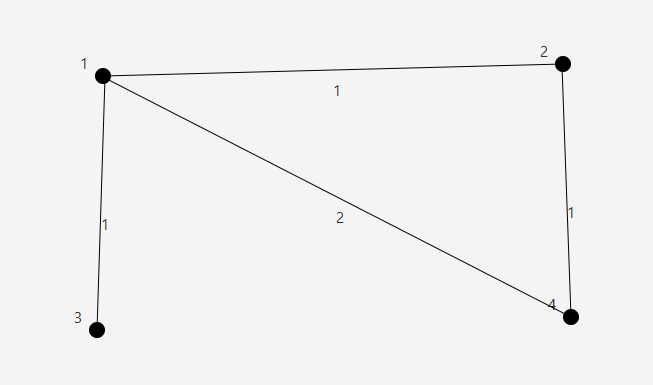
You need to display the maximum length of one cable in terms of connection. Also you need to highlight the edges that will be used in connections.

**Decision:**

For this weighted graph we need to find the spanning tree such that the max weights of the edges are minimal.

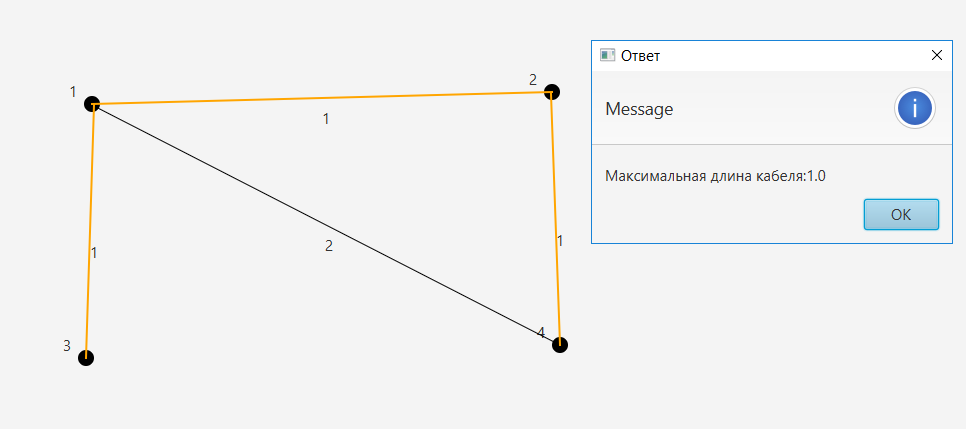
**Implement:**

**Input data:**



Pic. 3 – Input data for Kruskal`s alg.

**Output data:**



Pic.4 – Output data of Kruskal`s alg.

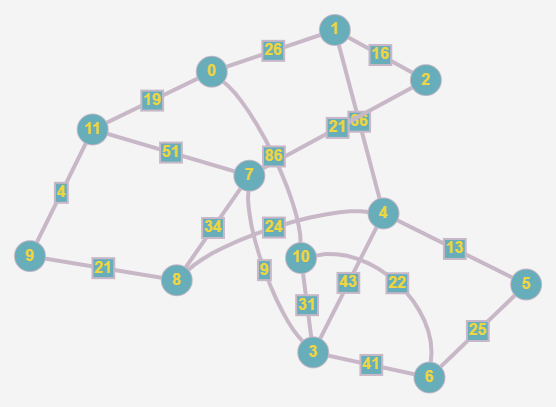
Dijkstra, Floyd-Warshall and Ford-Bellman

**Task:**

The company engaged in the transportation of perishable goods was given the task of delivering goods from Stavropol to Budyonnovsk, and there are several ways which it is possible to deliver that. The distance between the city of Stavropol and the village of K. is 26 kilometers, between the city of Stavropol and the village of P. is 19 kilometers, between the city of Stavropol and the village of R. is 86 kilometers. Between villages K. and D. - 16 kilometers, between villages K. and L. - 66 kilometers. Between the village of P. and the city of N. it is 4 kilometers, between the villages of P. and B. - 51 kilometers. Between villages D. and V. - 21 kilometers. Between the city of N. and the village of M. - 21 kilometers. Between the villages of M. and L. - 24 kilometers, between the villages of M. and V. - 34 kilometers. Between the villages of L. and A. - 13 kilometers, between the villages of L. and J. - 43 kilometers. Between villages A. and B. - 25 kilometers. Between villages Zh. And R. - 31 kilometers, between villages Zh. And B. - 44 kilometers. Between villages B. and R. - 22 kilometers. Between villages V. and Zh. - 9 kilometers. It is necessary to find the shortest route from Stavropol to Budennovsk.

**Decision:**

Let`s visualize this task as a graph.



Pic. 5 – Graph

Now we see, that node 0 is Stavropol, and node 6 is Budyonnovsk. Using these 3 algorithms we need to find the minimal way between these two nodes.

**Note:**

The answers that we received must be equals!

**Implement:**

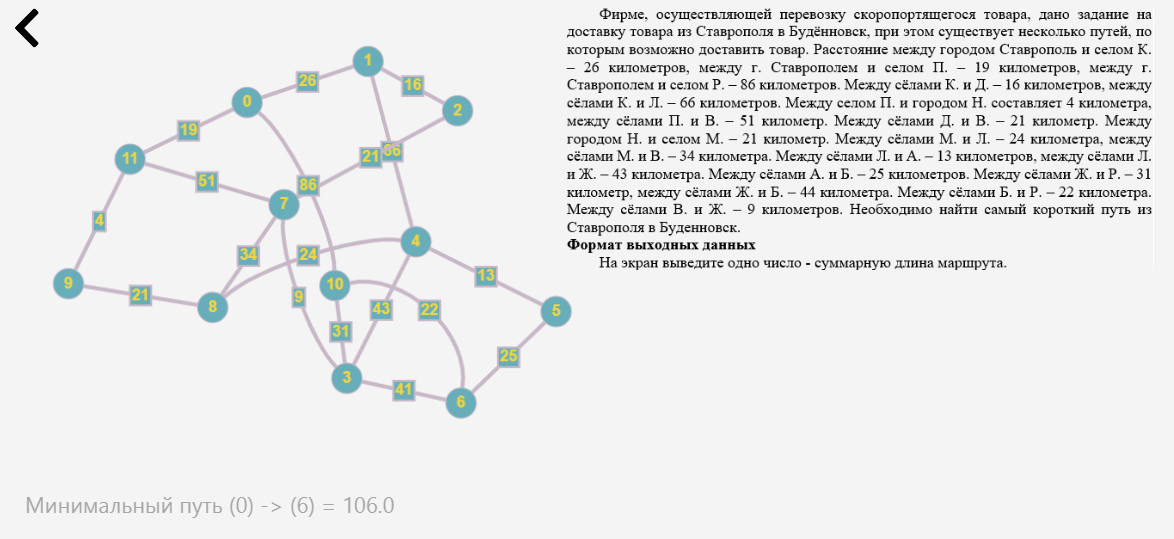
**NOTE:**

As the answers is the same for each algorithm, only the dijkstra algorithm will be presented.

**Input data:**

The adjacency matrix.

**Output data:**



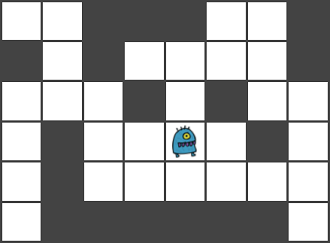
Pic.6 – Implement of Dijkstra alg.

Ford-Fulkerson

**Task:**

Given a field of size N × N, some cells of the field are shaded. In one of the unfilled cells is the Minotaur, it can only walk along unfilled cells (from the current cell, it can only go to a neighboring unfilled cell). What is the minimum number of cells that need to be painted over so that the Minotaur cannot get out of the field (reach any extreme cell)?

**Example:**



**Solution and proof of correctness**

**Theorem**:

The minimum number of cells to be colored is equal to the maximum number of cell-disjoint paths from the position of the Minotaur to the verge cells of the field.

**Evidence**:

Obviously, the answer is no more than the number of all paths from the Minotaur to the extreme cells. Let's make even more severe inequality: the answer is no more than the maximum number of cell-disjoint paths, because if you take any 2 intersecting paths and paint over the cell in the position where they intersect, then the exit from the field is blocked immediately along 2 of these paths. On the other hand, if you paint over a cell on one of the paths, then only this path is blocked, because cell-disjoint paths were taken. So the answer is no less than the number of such paths. One more inequality: the answer can`t be more than 4, because we can paint the neighboring cells of Minotaur cell (count of neighboring cells for each cell can`t be more than 4, because square has only 4 edges).

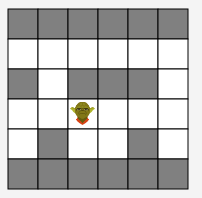
**Transition to the network**

Consider a network in which nodes of graphs will be all unpainted cells, connect adjacent unfilled neighboring cells with oriented edges with a weight of 1. As a source, we take the node which correspond the Minotaur cell. Add another vertex to the graph - the drain, add edges from the vertices corresponding to the verge cells of the field, to the drain with a weight of 1.

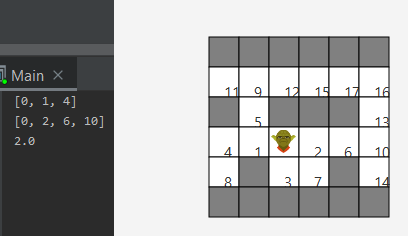
Using the Ford-Fulkerson algorithm, we find the maximum flow in the network. According to the decomposition theorem, finding the maximum flow with weight 1 of each edges is equivalent to finding the maximum number of paths from source to drain. Those. the required answer to the problem is equal to the maximum flow.

**Implement:**

**Input data:**



**Output data:**



Where integer numbers on the map is node number in graph. Integers massive is the maximum flow on the graph. And the last double number is the count of cells that we need to paint.