

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Departamentul Ingineria Software și Automatică

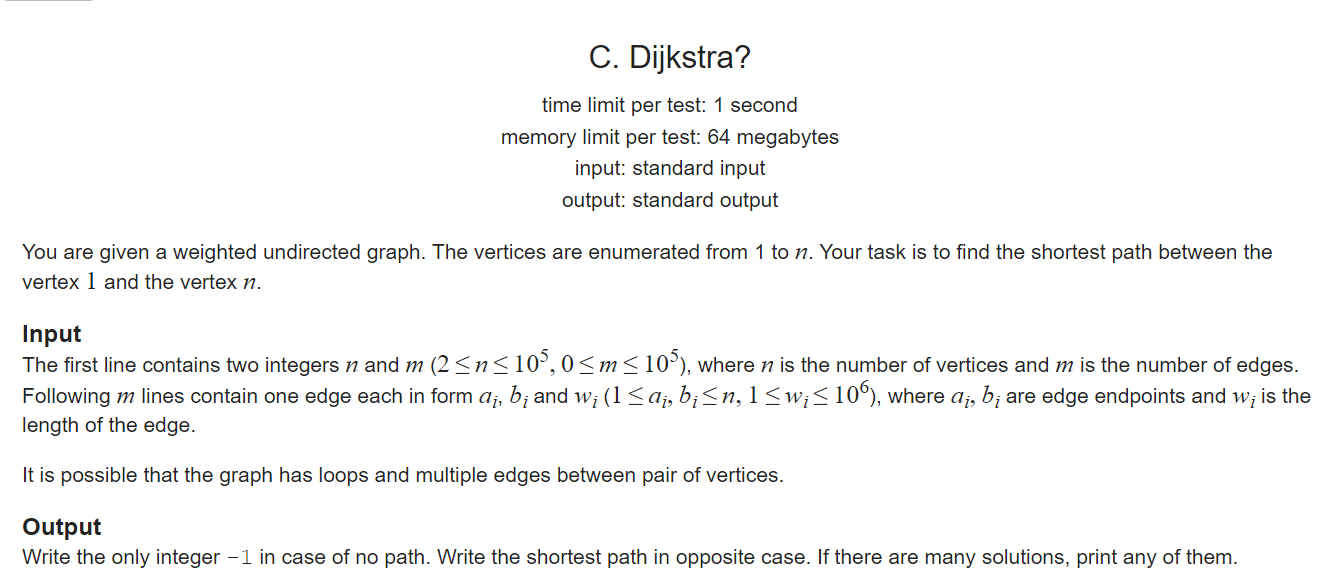
RAPORT

Lucrare de laborator nr.1

Structuri de date si algoritmi

A efectuat: st. gr.FAF-212 Lupascu Felicia

A verificat: dr. conf. Univ S. Corlat



Dijkstra's algorithm finds the shortest path between any two graph vertices. It differs from the minimum spanning tree because the shortest distance between two vertices might not include all the vertices of the graph.

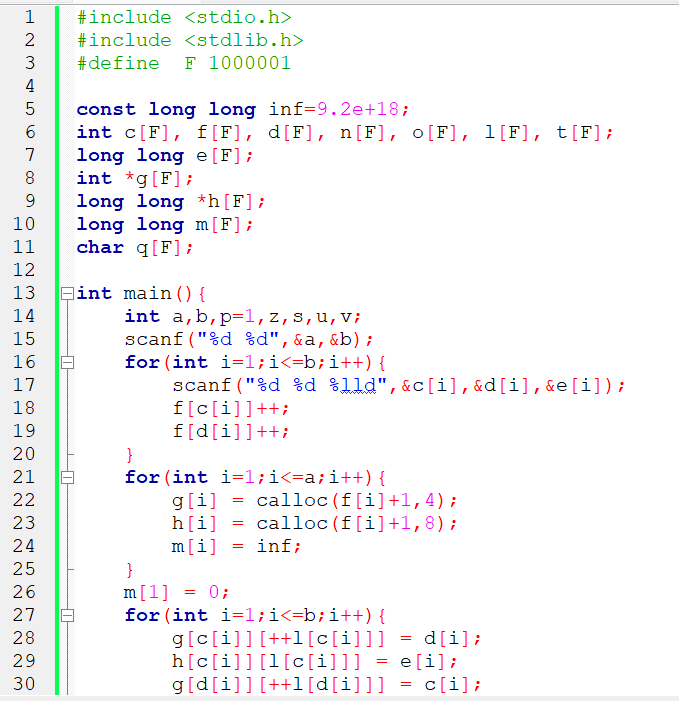
I use Dijkstra to find the shortest path value, and the shortest path should be recorded here. Here's how to record the path:  
1. Using the Dijkstra algorithm to find the shortest path. The relaxation operation is actually a process of continuously selecting the shortest path. Each relaxation is a process of " **selecting a path shorter than the current path** ". In the end, The end of the relaxation operation means that the selection of the shortest path is determined.  
2. Because **the choice of the path** exists in the relaxation operation, we should record the path in the relaxation operation. In the Dijkstra algorithm, each relaxation operation uses a " **point with the shortest path determined** " to relax other points, that is, the edge used for relaxation each time point from the point with the shortest path to the "shortest **path** ".

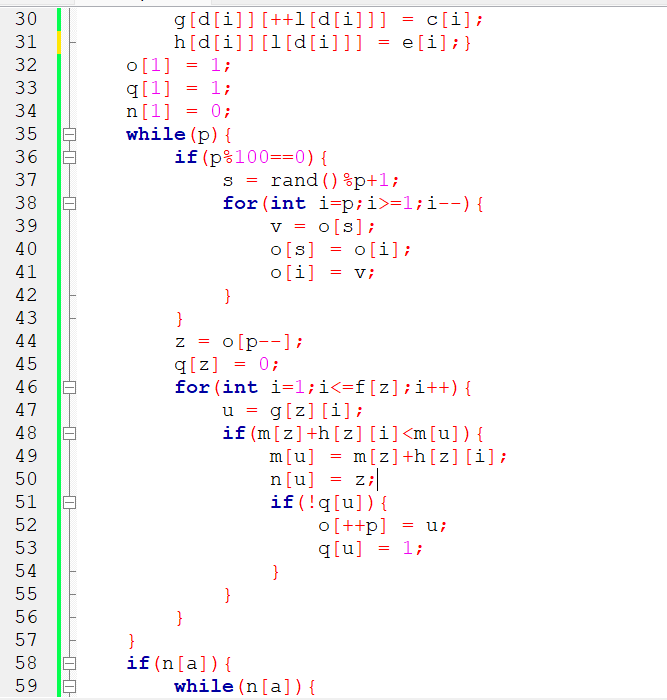
At each step:

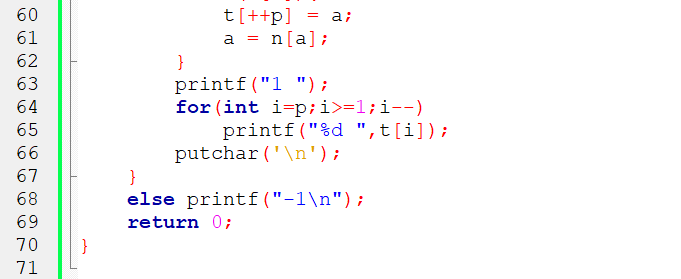
1. Pick the closest unknown vertex
2. Add it to known vertices
3. Update distances

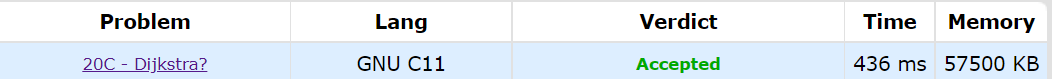
The algorithm works by building a set of nodes that have a minimum distance from the source.

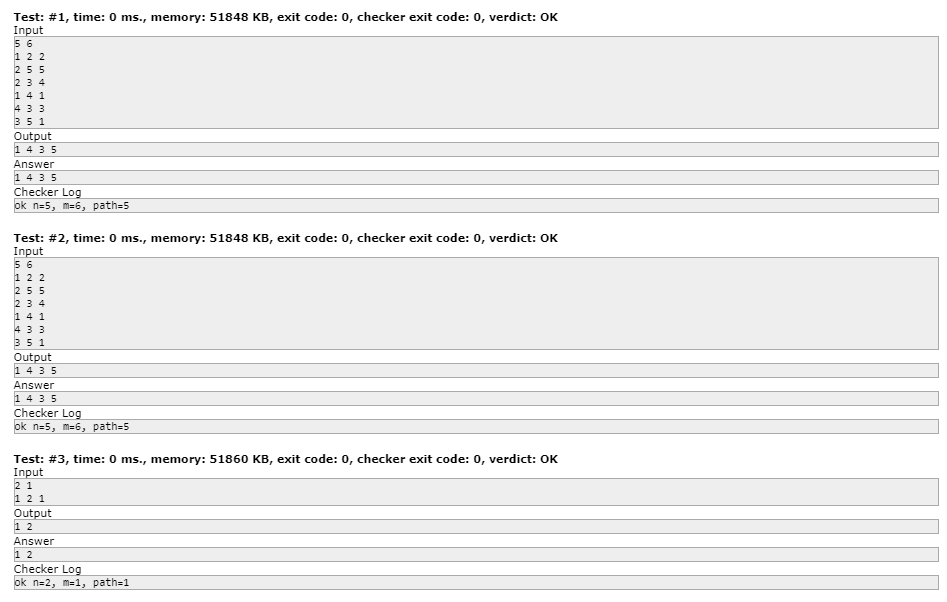
**No Heaps Implementation**



















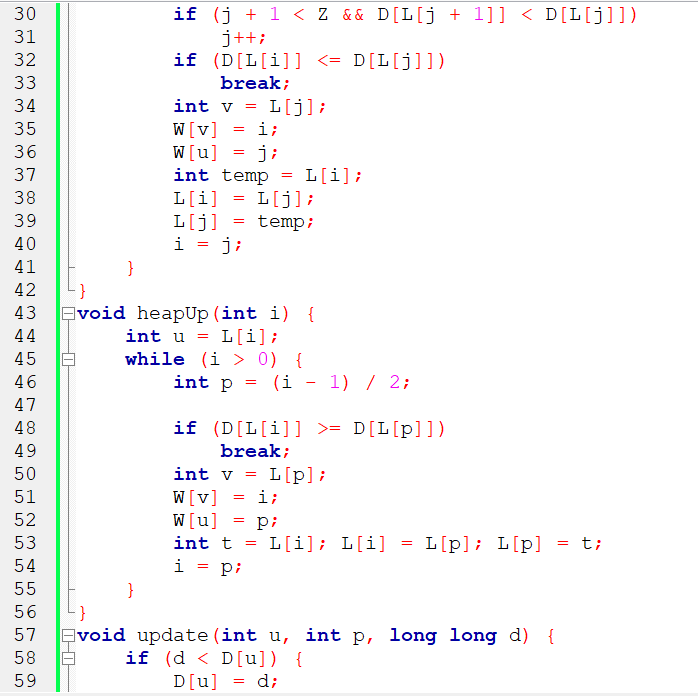
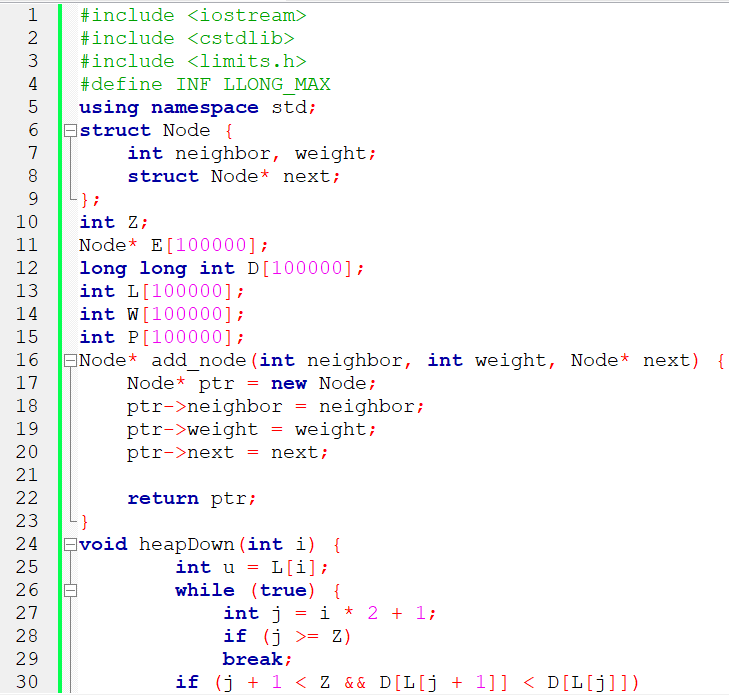


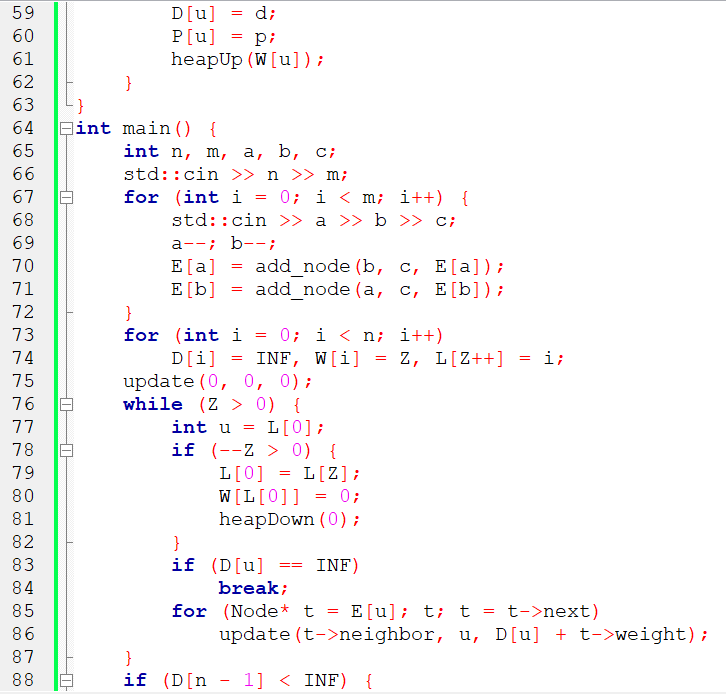
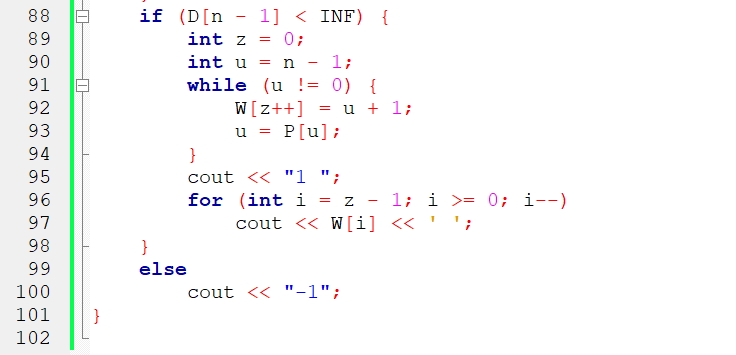




***2.***

















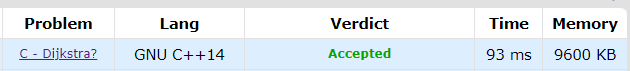


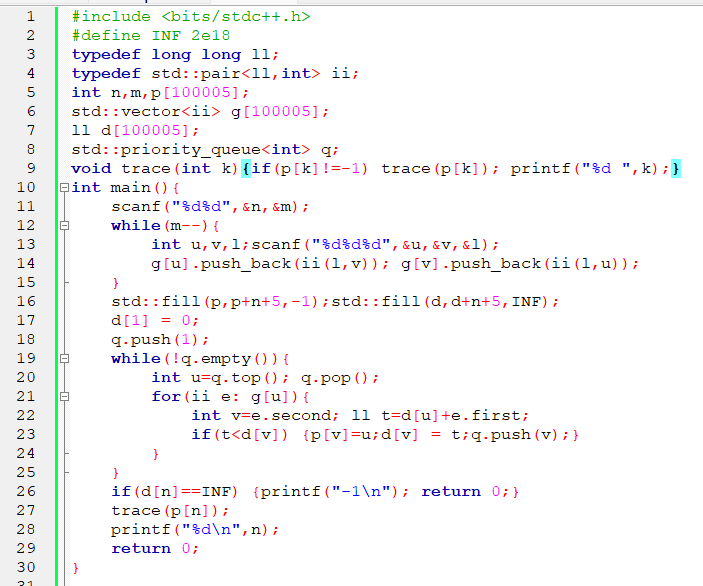






**3.**





**Test: #1, time: 0 ms., memory: 2344 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #2, time: 0 ms., memory: 2344 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #3, time: 0 ms., memory: 2352 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #4, time: 0 ms., memory: 2344 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #5, time: 15 ms., memory: 2348 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #6, time: 0 ms., memory: 2348 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #7, time: 0 ms., memory: 2344 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #8, time: 0 ms., memory: 2344 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #9, time: 0 ms., memory: 2352 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #10, time: 0 ms., memory: 2568 KB, exit code: 0, checker exit code: 0, verdict: OK**

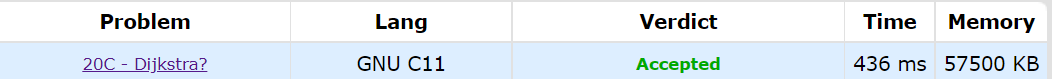
….

**Test: #32, time: 93 ms., memory: 8764 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Test: #33, time: 78 ms., memory: 8416 KB, exit code: 0, checker exit code: 0, verdict: OK**

**Conclusions:**

***For the first case:***

**

***For the second case:***

**

***For the third case:***

