

Graph algorithms - practical work no. 3

Due: week 9-10.

Note: computing the minimum cost walk (as some kind of a list of vertices) is a mandatory part of the assignment.

Solve the assigned problem from those below. Additionally, you may solve one or more of the bonus problems at the end of the page. Use the abstract data type created for lab. 1; modify it if necessary.

Assigned problems

1. Write a program that, given a graph with positive costs and two vertices, finds a lowest cost walk between the given vertices, using the Dijkstra algorithm.
2. Write a program that, given a graph with positive costs and two vertices, finds a lowest cost walk between the given vertices, using a "backwards" Dijkstra algorithm (Dijkstra algorithm that searches backwards, from the ending vertex).
3. Write a program that, given a graph with costs and two vertices, finds a lowest cost walk between the given vertices, or prints a message if there are negative cost cycles accessible from the starting vertex. The program will use a matrix defined as $d[x,k]=\text{the cost of the lowest cost walk from } s \text{ to } x \text{ and of length at most } k$, where s is the starting vertex.
4. Write a program that, given a graph with costs and two vertices, finds a lowest cost walk between the given vertices, or prints a message if there are negative cost cycles accessible from the starting vertex. The program will use a matrix defined as $d[x,k]=\text{the cost of the lowest cost walk from } s \text{ to } x \text{ and of length equal to } k$, where s is the starting vertex.
5. Write a program that, given a graph with costs and two vertices, finds a lowest cost walk between the given vertices, or prints a message if there are negative cost cycles accessible from the starting vertex. The program will use the Ford's algorithm.
6. Write a program that, given a graph with costs and two vertices, finds a lowest cost walk between the given vertices, or prints a message if there are negative cost cycles in the graph. The program shall use the matrix multiplication algorithm.
7. Write a program that, given a graph with costs that has no negative cost cycles and two vertices, finds a lowest cost walk between the given vertices. The program shall use the Floyd-Warshall algorithm.

Optional, "bonus" problems

- 1B. Write a program that, given a graph with costs, having no negative cost cycles, and a pair of vertices, finds the number of distinct walks of minimum cost between the given vertices.
- 2B. Write a program that, given a graph that has no cycles (a directed acyclic graph, DAG) and a pair of vertices, finds the number of distinct walks between the given vertices.
- 3B. Use a lowest cost path algorithm and a new implementation of the interface from lab 1 to solve the *bridge and torch* problem: A number of people (up to 20) must cross a bridge, at night. They have a torch that provides only enough light for at most two people to cross together. For each person, we are given the time needed to cross the bridge alone. If two people cross together, they cross at the pace of the slowest one. Find a solution for all to cross the river, using as little time as possible.