

Algorithms

Lab 7 (8%)

Topics: Algorithms on weighted graphs

Textbook: Cormen Introduction to Algorithms 3rd ed.

Problem 01 (2%)

Minimum Spanning Tree, Prim's Algorithm

1. Read Cormen 23.1, 23.2, Appendix B.4

2. Use Prim's algorithm to implement following C++ program:

Given a number of edges of undirected weighted connected graph and list of graph's edges in following format: <char> <char> <int>. Characters <char> represent vertices of an edge and integer represents edge's weight. Program has to output all edges of some MST. Use simplest way to find minimum-weight edge on each step of Prim's algorithm.

Sample Input:

```
10
a b 9
a c 9
a f 8
a j 18
b c 3
b e 6
c e 4
c d 2
c f 9
d e 2
d f 8
d g 9
e g 9
f g 7
f i 9
f j 10
g h 4
g i 5
i h 1
i j 3
h j 4
```

Sample Output:

```
a f 8
f g 7
g h 4
h i 1
i j 3
f d 8
d c 2
d e 2
c b 3
```

Problem 02 (2%)

Single-Source Shortest Path. Dijkstra's Algorithm

1. Read Cormen 24.1, 24.3, Appendix B.4
2. Use Dijkstra's algorithm to implement following C++ program:

Given an undirected weighted graph in following format:

<number of vertices>

<list of edges>

and

<source vertex>

<destination vertex>

Note: list of edges ends by 3 zeros. Each edge are represented by 3 integer values:

<vertex> <vertex> <weight>

Vertices are labeled by integer values starting from 1.

Program has to output shortest path from source vertex to destination vertex (see sample output).

Use simplest way to find "closest" vertex on each step of Dijkstra's algorithm

Sample Input:

```
9
1 2 2
1 3 4
1 4 3
2 5 10
2 3 1
3 5 12
3 4 5
4 5 7
4 6 9
4 7 13
5 6 5
5 8 12
6 7 4
6 8 14
6 9 16
7 9 10
8 9 2
0 0 0
1
9
```

Sample Output:

```
1 4 5 8 9
```

Problem 03 (2%)

Min-Heap

1. Read Cormen. Chapter 6.
2. Implement Min-Heap in C++ program which has to read arbitrary integer values from standard input until the end of input and output them in non-decreasing order. Heap must be represented by a vector of integer values. Program has to have following functions:

```
const int& top(const vector<int>& heap)
void pop(vector<int>& heap)
void push(vector<int>& heap, int x)
```

Problem 04 (2%)

Single-Source Shortest Paths, Bellman-Ford Algorithm

3. Read Cormen 24.1, Appendix B.4

4. Use Bellman-Ford algorithm to implement following C++ program:

Given a number of edges of directed weighted connected graph and list of graph's edges in following format: <char> <char> <int>. Characters represent vertices of an edge and integer represents edge's weight. Input ends by <source vertex>. Program has to output shortest paths to all vertices from <source vertex>. If a given graph has a negative loop program has to print corresponding message.

Sample Input:

```
9
A B 10
B C 2
C B -1
D C 4
E C 4
B E -3
A F 3
F E 3
E D 1
A
```

Sample Output:

```
A: A
B: A F E C B
C: A F E C
D: A F E D
E: A F E
F: A F
```