Politechnika Śląska
Wydział Informatyki, Elektroniki i Informatyki

Fundamentals of Computer Programming

«Spanning Tree»

author	Dawid Bogocz
instructor	dr inż. Pablo Ribalta Lorenzo
year	2020/2021
lab group	Thursday, 11:30 – 16:00
deadline	2020-11-08

1 Project's topic

Implement a program that finds a minimal spanning tree in a weighted graph. A graph is stored in a text file as a sequence of comma separated triples. Each triple is parenthesised and holds three comma separated values that represent two nodes that define an edge and a weight of an edge. The program elaborates a minimal spanning tree and stores it in a text file in the same format as the input file. The program is run in command line with switches: `-i` input file name
`-o` output file name

2 Analysis of the task

The task focuses on finding a minimal spanning tree in a weighted graph.

2.1 Data structures

Vector is used to store values read from an input file. Each edge is represented with a pair of weight and another pair of two nodes.

2.2 Algorithms

The Program uses Kruskal's Algorith to find Minimal spanning tree of an undirected edge-weighted graph. It is a greedy algorithm, in each step it adds the next lowest-weight edge that will not form a cycle to the minimum spanning tree.

3 External specification

This is a command line program.The program requires names of input and output files. Put input file name after `-i` switch and output file name after `-o` switch, eg:

```
program -i input-file -o output-file
program -o output-file -i input-file
```

Both files are text files. The switches may be provided in any order. The program called with no parameters or with parameter `-h` prints help (a short manual).

Program call

```
program
program -h
```

prints a short manual. Program called with incorrect parameters prints an error message:

```
Incorrect paramters!
```

and prints help.

Incorrect file names are detected and cause a message:

```
File <name-of-file> not found!
```

4 Internal specification

4.1 Program overview

The main function calls read that opens a file stream, reads numbers from the stream, and inserts values from input file into a vector. If the input file is't found, an appropriate message is printed. Then main function callls MST that using Kruskal's aghorithm finds Minimal Spanning Tree. Finally the program prints numbers into an output file.

4.2 Description of types and functions

Description of types and functions is moved to the appendix.

5 Testing

The program has been tested with test.txt file. Incorrect path to file is detected and an error message is printed. An empty input file causes failure – output file is not created.

6 Conclusions

The program implements Kruskal's algorith that works with disjoint set data structure. The most challenging task is creating data structure that represents weighted graph and is easy to navigate.

Appendix Description of types and functions

My Project

Generated by Doxygen 1.8.20

1 Class Index 1

1.1 Class List . . . 1

2 File Index 3

2.1 File List . . . 3

3 Class Documentation 5

3.1 Graph Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Constructor & Destructor Documentation . . . 5

3.1.2.1 Graph() . . . 5

3.1.3 Member Function Documentation . . . 6

3.1.3.1 find() . . . 6

3.1.3.2 MST() . . . 6

3.1.3.3 print() . . . 6

3.1.3.4 read() . . . 6

3.1.3.5 unite() . . . 7

4 File Documentation 9

4.1 functions.h File Reference . . . 9

4.1.1 Function Documentation . . . 9

4.1.1.1 count() . . . 9

4.1.1.2 usage() . . . 10

4.2 graph.txt File Reference . . . 10

4.3 Source1.cpp File Reference . . . 10

4.3.1 Function Documentation . . . 10

4.3.1.1 main() . . . 10

4.4 test.txt File Reference . . . 10

Index 11

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Graph 5

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

functions.h 9
Source1.cpp 10

Chapter 3

Class Documentation

3.1 Graph Class Reference

`#include <functions.h>`

Public Member Functions

- Graph (int nodes)
- void read (std::string input)
- int find (int x)
- void unite (int a, int b)
- void MST ()
- void print (std::string outpout)

3.1.1 Detailed Description

Definition at line 9 of file functions.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Graph()

```
Graph::Graph (
            int nodes )
```

Function that creates N disjoint sets.

Definition at line 64 of file functions.h.

3.1.3 Member Function Documentation

3.1.3.1 find()

```
int Graph::find (
            int x )
```

Function that finds the root of the set in which element x belongs.

Definition at line 113 of file functions.h.

3.1.3.2 MST()

```
void Graph::MST ( )
```

Function that finds Minimal Spanning Tree using Kruskal's algorithm and disjoint-set.

Parameters

a	represents one of two nodes connected with weighted edge.
b	represents one of two nodes connected with weighted edge.
weight	represents weighted edge.

Definition at line 127 of file functions.h.

3.1.3.3 print()

```
void Graph::print (
            std::string outpout )
```

Function that prints result of the program into output file.

Definition at line 142 of file functions.h.

3.1.3.4 read()

```
void Graph::read (
            std::string input )
```

Function that reads input file and puts values stored in it into vector edges.

Parameters

line	stores one line taken from input file.
rows	stores number of rows from vector contents that represent number of edges.
part	stores a part of line and is used to extract numbers form parentheses.
x	represents one of two nodes connected with weighted edge.
y	represents one of two nodes connected with weighted edge.
w	represents weighted edge.

Definition at line 71 of file functions.h.

3.1.3.5 unite()

```
void Graph::unite (
            int a,
            int b )
```

Function that unites of two subsets .

Parameters

a	represents first subset.
b	represents second subset.

Definition at line 121 of file functions.h.

The documentation for this class was generated from the following file:

- functions.h

Chapter 4

File Documentation

4.1 functions.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <algorithm>
#include <set>
```

Classes

- class Graph

Functions

- int count (std::string input)
- void usage ()

4.1.1 Function Documentation

4.1.1.1 count()

```
int count (
            std::string input )
```

Function that finds how many nodes are in the weighted graph.

Parameters

line	stores one line taken from input file.
unique	stores unique nodes (one node can be called multiple times in input file to create a graph).

Definition at line 160 of file functions.h.

4.1.1.2 usage()

```
void usage ( )
```

Function that prints a help message on how to use flags in command line.

Definition at line 188 of file functions.h.

4.2 graph.txt File Reference

4.3 Source1.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <algorithm>
#include <set>
#include "functions.h"
```

Functions

- int main (int argc, char ∗argv[ ])

4.3.1 Function Documentation

4.3.1.1 main()

```
int main (
            int argc,
            char * argv[] )
```

Definition at line 11 of file Source1.cpp.

4.4 test.txt File Reference

Index

count
    functions.h, 9

find
    Graph, 6
functions.h, 9
    count, 9
    usage, 10

Graph, 5
    find, 6
    Graph, 5
    MST, 6
    print, 6
    read, 6
    unite, 7
graph.txt, 10

main
    Source1.cpp, 10
MST
    Graph, 6

print
    Graph, 6

read
    Graph, 6

Source1.cpp, 10
    main, 10

test.txt, 10

unite
    Graph, 7
usage
    functions.h, 10