

## Dijkstra Project

Generated by Doxygen 1.8.20



<b>1 File Index</b>	<b>1</b>
1.1 File List	1
<b>2 File Documentation</b>	<b>3</b>
2.1 dijkstraAlgorithm.cpp File Reference	3
2.1.1 Function Documentation	3
2.1.1.1 dijkstra()	3
2.1.1.2 findPathsFromVertices()	3
2.2 dijkstraAlgorithm.h File Reference	4
2.2.1 Typedef Documentation	4
2.2.1.1 adjPair	4
2.2.2 Function Documentation	4
2.2.2.1 dijkstra()	4
2.2.2.2 findPathsFromVertices()	5
2.3 graph.txt File Reference	5
2.4 graphCreation.cpp File Reference	5
2.4.1 Function Documentation	5
2.4.1.1 addToGraph()	6
2.4.1.2 createGraphBasedOnFile()	6
2.5 graphCreation.h File Reference	6
2.5.1 Typedef Documentation	6
2.5.1.1 adjPair	6
2.5.2 Function Documentation	7
2.5.2.1 addToGraph()	7
2.5.2.2 createGraphBasedOnFile()	7
2.6 main.cpp File Reference	7
2.6.1 Function Documentation	8
2.6.1.1 main()	8
2.7 miscellaneous.cpp File Reference	8
2.7.1 Function Documentation	8
2.7.1.1 reverseStr()	8
2.8 miscellaneous.h File Reference	8
2.8.1 Function Documentation	9
2.8.1.1 reverseStr()	9
2.9 parametersHandlingF.cpp File Reference	9
2.9.1 Function Documentation	9
2.9.1.1 assignIOfileNames()	9
2.9.1.2 assignOperations()	9
2.9.1.3 checkFlag()	10
2.10 parametersHandlingF.h File Reference	10
2.10.1 Function Documentation	10
2.10.1.1 assignIOfileNames()	10

2.10.1.2 assignOperations()	11
2.10.1.3 checkFlag()	11
2.11 test.txt File Reference	11
2.12 vertices.txt File Reference	11
<b>Index</b>	<b>13</b>

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<b>dijkstraAlgorithm.cpp</b>	3
<b>dijkstraAlgorithm.h</b>	4
<b>graphCreation.cpp</b>	5
<b>graphCreation.h</b>	6
<b>main.cpp</b>	7
<b>miscellaneous.cpp</b>	8
<b>miscellaneous.h</b>	8
<b>parametersHandlingF.cpp</b>	9
<b>parametersHandlingF.h</b>	10



## Chapter 2

# File Documentation

## 2.1 dijkstraAlgorithm.cpp File Reference

```
#include "dijkstraAlgorithm.h"  
#include "miscellaneous.h"
```

### Functions

- void **dijkstra** (vector< **adjPair** > adj[], vector< int > &vertices, int findFor, long long unsigned int adjSize, string &output)
- bool **findPathsFromVertices** (vector< **adjPair** > adj[], vector< int > &vertices, long long unsigned int adjSize, string sourceFileName, string resultFileName)

### 2.1.1 Function Documentation

#### 2.1.1.1 dijkstra()

```
void dijkstra (  
    vector< adjPair > adj[],  
    vector< int > & vertices,  
    int findFor,  
    long long unsigned int adjSize,  
    string & output )
```

#### 2.1.1.2 findPathsFromVertices()

```
bool findPathsFromVertices (  
    vector< adjPair > adj[],  
    vector< int > & vertices,  
    long long unsigned int adjSize,  
    string sourceFileName,  
    string resultFileName )
```

## 2.2 dijkstraAlgorithm.h File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <string>
#include <queue>
#include <limits>
```

### Typedefs

- typedef std::pair< int, double > **adjPair**

### Functions

- void **dijkstra** (std::vector< **adjPair** > adj[], std::vector< int > &vertices, int findFor, long long unsigned int adjSize, std::string &output)
- bool **findPathsFromVertices** (std::vector< **adjPair** > adj[], std::vector< int > &vertices, long long unsigned int adjSize, std::string sourceFileName, std::string resultFileName)

### 2.2.1 Typedef Documentation

#### 2.2.1.1 adjPair

```
typedef std::pair<int, double> adjPair
```

### 2.2.2 Function Documentation

#### 2.2.2.1 dijkstra()

```
void dijkstra (
    std::vector< adjPair > adj[],
    std::vector< int > & vertices,
    int findFor,
    long long unsigned int adjSize,
    std::string & output )
```

Function that implements Dijkstra's algorithm and shows the path.

Dijkstra's algorithm (or Dijkstra's Shortest Path First) is an algorithm for finding the shortest paths between nodes in a graph.(src [https://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm))



## Parameters

<i>adj[]</i>	- vector of pairs of number of vertex and weight towards adjacent vertex (graph)
<i>vertices</i>	- vector of vertices
<i>findFor</i>	- index of vertex to search from
<i>adjSize</i>	- size of vector adj
<i>output</i>	- string containing shortest paths and their distances

## 2.2.2.2 findPathsFromVertices()

```
bool findPathsFromVertices (
    std::vector< adjPair > adj[],
    std::vector< int > & vertices,
    long long unsigned int adjSize,
    std::string sourceFileName,
    std::string resultFileName )
```

Function that reads vertices from a file to search from and uses dijkstra function on an existing graph to write result to a file.

## Parameters

<i>adj[]</i>	- vector of pairs of number of vertex and weight towards adjacent vertex (graph)
<i>vertices</i>	- vector of vertices
<i>adjSize</i>	- size of vector adj
<i>sourceFileName</i>	- path to file containing vertices to search from
<i>resultFileName</i>	- path to file that should store the result

## 2.3 graph.txt File Reference

## 2.4 graphCreation.cpp File Reference

```
#include "graphCreation.h"
```

## Functions

- void **addToGraph** (vector< **adjPair** > adj[], vector< int > &vertices, int u, int v, double weight)
- bool **createGraphBasedOnFile** (vector< **adjPair** > adj[], vector< int > &vertices, string sourceFileName)

## 2.4.1 Function Documentation

### 2.4.1.1 addToGraph()

```
void addToGraph (
    vector< adjPair > adj[],
    vector< int > & vertices,
    int u,
    int v,
    double weight )
```

### 2.4.1.2 createGraphBasedOnFile()

```
bool createGraphBasedOnFile (
    vector< adjPair > adj[],
    vector< int > & vertices,
    string sourceFileName )
```

## 2.5 graphCreation.h File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
```

### Typedefs

- typedef std::pair< int, double > **adjPair**

### Functions

- void **addToGraph** (std::vector< **adjPair** > adj[], std::vector< int > &vertices, int u, int v, double weight)
- bool **createGraphBasedOnFile** (std::vector< **adjPair** > adj[], std::vector< int > &vertices, std::string sourceFileName)

### 2.5.1 Typedef Documentation

#### 2.5.1.1 adjPair

```
typedef std::pair<int, double> adjPair
```

## 2.5.2 Function Documentation

### 2.5.2.1 addToGraph()

```
void addToGraph (
    std::vector< adjPair > adj[],
    std::vector< int > & vertices,
    int u,
    int v,
    double weight )
```

Function to add vertices and edges to a graph.

#### Parameters

<i>adj[]</i>	- vector of pairs of number of vertex and weight towards adjacent vertex (graph)
<i>vertices</i>	- vector of vertices
<i>u</i>	- first vertex
<i>v</i>	- second vertex
<i>weight</i>	- edge weight first -> second

### 2.5.2.2 createGraphBasedOnFile()

```
bool createGraphBasedOnFile (
    std::vector< adjPair > adj[],
    std::vector< int > & vertices,
    std::string sourceFileName )
```

Function to create graph based on a given file.

#### Parameters

<i>adj[]</i>	- vector of pairs of number of vertex and weight towards adjacent vertex (graph)
<i>vertices</i>	- vector of vertices
<i>sourceFileName</i>	- path to file that is used to generate a graph

## 2.6 main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <queue>
```

```
#include <limits>
#include "parametersHandlingF.h"
#include "graphCreation.h"
#include "miscellaneous.h"
#include "dijkstraAlgorithm.h"
```

## Functions

- int **main** (int argc, char \*argv[])

### 2.6.1 Function Documentation

#### 2.6.1.1 main()

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

## 2.7 miscellaneous.cpp File Reference

```
#include <vector>
#include <string>
```

## Functions

- void **reverseStr** (string &str)

### 2.7.1 Function Documentation

#### 2.7.1.1 reverseStr()

```
void reverseStr (
    string & str )
```

## 2.8 miscellaneous.h File Reference

## Functions

- void **reverseStr** (std::string &str)

## 2.8.1 Function Documentation

### 2.8.1.1 reverseStr()

```
void reverseStr (
    std::string & str )
```

Function that reverses a string

#### Parameters

<i>str</i>	- string to be reversed
------------	-------------------------

## 2.9 parametersHandlingF.cpp File Reference

```
#include "parametersHandlingF.h"
```

### Functions

- void **assignOperations** (char \*fName, string &nameVar, string errorMessage)
- void **checkFlag** (char \*flag, char \*fName, string &inGraph, string &inVertices, string &outResult)
- bool **assignIOfileNames** (int argc, char \*argv[], string &inGraph, string &inVertices, string &outResult)

## 2.9.1 Function Documentation

### 2.9.1.1 assignIOfileNames()

```
bool assignIOfileNames (
    int argc,
    char * argv[],
    string & inGraph,
    string & inVertices,
    string & outResult )
```

### 2.9.1.2 assignOperations()

```
void assignOperations (
    char * fName,
    string & nameVar,
    string errorMessage )
```

### 2.9.1.3 checkFlag()

```
void checkFlag (
    char * flag,
    char * fName,
    string & inGraph,
    string & inVertices,
    string & outResult )
```

## 2.10 parametersHandlingF.h File Reference

```
#include <iostream>
#include <string>
```

### Functions

- void **assignOperations** (char \*fName, std::string &nameVar, std::string errorMessage)
- void **checkFlag** (char \*flag, char \*fName, std::string &inGraph, std::string &inVertices, std::string &outResult)
- bool **assignIOfileNames** (int argc, char \*argv[], std::string &inGraph, std::string &inVertices, std::string &outResult)

### 2.10.1 Function Documentation

#### 2.10.1.1 assignIOfileNames()

```
bool assignIOfileNames (
    int argc,
    char * argv[],
    std::string & inGraph,
    std::string & inVertices,
    std::string & outResult )
```

Function that handles program's arguments. If none are specified, it prints explanation about program's usage

#### Parameters

<i>argc</i>	- number of arguments
<i>argv</i> []	- array of arguments
<i>inGraph</i>	- string for graph's input file name
<i>inVertices</i>	- string for vertices's input file name
<i>outResult</i>	- string for result's output file name

### 2.10.1.2 assignOperations()

```
void assignOperations (
    char * fName,
    std::string & nameVar,
    std::string errorMessage )
```

Function to assign path to file to a given string. If there is no path (nullptr), function prints error message.

#### Parameters

<i>fName</i>	- path to a file
<i>nameVar</i>	- variable to store path if it exists
<i>erroeMessage</i>	- a message to be printed in case there is no path specified

### 2.10.1.3 checkFlag()

```
void checkFlag (
    char * flag,
    char * fName,
    std::string & inGraph,
    std::string & inVertices,
    std::string & outResult )
```

Function that checks given flags and processes them accordingly.

#### Parameters

<i>flag</i>	- variable that defines what to do with fName
<i>fName</i>	- variable that stores name of one of the files
<i>inGraph</i>	- string for graph's input file name
<i>inVertices</i>	- string for vertices's input file name
<i>outResult</i>	- string for result's output file name

## 2.11 test.txt File Reference

## 2.12 vertices.txt File Reference





# Index

- addToGraph
  - graphCreation.cpp, 5
  - graphCreation.h, 7
- adjPair
  - dijkstraAlgorithm.h, 4
  - graphCreation.h, 6
- assignIOfileNames
  - parametersHandlingF.cpp, 9
  - parametersHandlingF.h, 10
- assignOperations
  - parametersHandlingF.cpp, 9
  - parametersHandlingF.h, 10
- checkFlag
  - parametersHandlingF.cpp, 9
  - parametersHandlingF.h, 11
- createGraphBasedOnFile
  - graphCreation.cpp, 6
  - graphCreation.h, 7
- dijkstra
  - dijkstraAlgorithm.cpp, 3
  - dijkstraAlgorithm.h, 4
- dijkstraAlgorithm.cpp, 3
  - dijkstra, 3
  - findPathsFromVertices, 3
- dijkstraAlgorithm.h, 4
  - adjPair, 4
  - dijkstra, 4
  - findPathsFromVertices, 5
- findPathsFromVertices
  - dijkstraAlgorithm.cpp, 3
  - dijkstraAlgorithm.h, 5
- graph.txt, 5
- graphCreation.cpp, 5
  - addToGraph, 5
  - createGraphBasedOnFile, 6
- graphCreation.h, 6
  - addToGraph, 7
  - adjPair, 6
  - createGraphBasedOnFile, 7
- main
  - main.cpp, 8
- main.cpp, 7
  - main, 8
- miscellaneous.cpp, 8
  - reverseStr, 8
- miscellaneous.h, 8
  - reverseStr, 9
- parametersHandlingF.cpp, 9
  - assignIOfileNames, 9
  - assignOperations, 9
  - checkFlag, 9
- parametersHandlingF.h, 10
  - assignIOfileNames, 10
  - assignOperations, 10
  - checkFlag, 11
- reverseStr
  - miscellaneous.cpp, 8
  - miscellaneous.h, 9
- test.txt, 11
- vertices.txt, 11