# My Project

Generated by Doxygen 1.8.2

Wed Nov 14 2012 08:15:21

# Contents

**1 The Prims Algorithm** **1**

**2 Class Index** **3**

2.1 Class List . . . . . 3

**3 File Index** **5**

3.1 File List . . . . . 5

**4 Class Documentation** **7**

4.1 keyComparison Struct Reference . . . . . 7

4.1.1 Member Function Documentation . . . . . 7

4.1.1.1 operator() . . . . . 7

4.2 prims Class Reference . . . . . 7

4.2.1 Member Function Documentation . . . . . 7

4.2.1.1 init . . . . . 7

4.3 vertex Struct Reference . . . . . 8

**5 File Documentation** **11**

5.1 main.cpp File Reference . . . . . 11

5.1.1 Detailed Description . . . . . 11

5.2 prims.cpp File Reference . . . . . 11

5.2.1 Detailed Description . . . . . 11

5.3 prims.h File Reference . . . . . 12

5.3.1 Detailed Description . . . . . 12

**Index** **12**

# Chapter 1

# The Prims Algorithm

**Problem Definition:**

∗Implement Prim’s MST algorithm using heaps. Your program must take two
le names as inputs from the user. The input
le will contain several test cases delimited by a $. Each test case is an edge-weighted graph (on the vertex set f1; 2; : : : ; ng) formatted as follows. The
rst line contains 2 integers n and m denoting the number of vertices and edges, respectively, in the graph. The next m lines represent the adjacency information. Each such line contains 3 integers i, j and wij that are interpreted as fi; j g being an edge of weight wij in the graph. The last line of input
le contains a $$ to indicate end of all test cases. The output
le should contain the list of edges of the MST (with associated cost) for each graph in the input. Note that the input graphs could be disconnected and in that case, no spanning tree exists.

**Algorithm:**

```
MST-PRIM(G,w,r)
    for each u V[G]
    do key[u] ←
    –[u]← NIL
    –key[r] ← 0
    Q ← V [G]
    while(q is not empty)
    –do u ← EXTRACT-MIN(Q)
    –for each vAdj[u]
    -—do if vQ and w(u,v) <key[v]
    ---—then [v]←u
    ---—key[v]← w (u,v)
```

O(ElogV)

**Format of contents in a input file:**

- 7 10
- 1 2 1
- 1 4 10
- 1 6 3
- 1 5 5
- 2 3 2
- 3 4 9

- 3 7 6
- 4 6 8
- 5 6 7
- 5 7 4
- $$

**Format of contents of expected output for the above input file:**

- edge 1,2 1
- edge 2,3 2
- edge 1,6 3
- edge 1,5 5
- edge 5,7 4
- edge 6,4 8
- $$

**Author**

Nitesh Bhargava

**Date**

6/11/12

# Chapter 2

# Class Index

## 2.1 Class List

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

keyComparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
prims . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
vertex . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

# Chapter 3

# File Index

## 3.1 File List

Here is a list of all documented files with brief descriptions:

main.cpp
Main execution file . . . . . 11
prims.cpp
Defintion file for prims.h . . . . . 11
prims.h
Header file for the prims algorithm . . . . . 12

# Chapter 4

# Class Documentation

## 4.1 keyComparison Struct Reference

### Public Member Functions

- bool operator() (vertex ∗a, vertex ∗b)

    *overloading the '()' operator so that we can use it to create min heap*

### 4.1.1 Member Function Documentation

#### 4.1.1.1 bool keyComparison::operator() ( vertex ∗ *a*, vertex ∗ *b* ) [inline]

overloading the '()' operator so that we can use it to create min heap

condition for min heap

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

- prims.cpp

## 4.2 prims Class Reference

### Public Member Functions

- void init ()

### 4.2.1 Member Function Documentation

#### 4.2.1.1 void prims::init ( )

init for every iteration of the input

**Algorithm:**

- MST-PRIM(G,w,r) -for each u V[G] -do key[u] ←
    - [u]← NIL
        - key[r] ← 0 -Q ← V [G] -while(q is not empty)
        - do u ← EXTRACT-MIN(Q)
        - foreachvAdj[u]

    * do ifvQandw(u,v) <key[v]
    * then[v]←u
    * key[v]←w(u,v)

O(ElogV) input number of vertices

input number of edges

array of vetex structure

array to maintain whether node is visited

adjacency matrix

initializing adjacency matrix to zero and all vertices to not visited

input edges from m to n with edge weight edgewght

filling adjacency matrix

adjacency matrix is symmetric

inserting the information of all vertices of graph into 'arr' array

value of that node

max possible value (positive infinity)

setting all parent to -1 means they have no parent ryt now

1 node is our starting point of mst

initialing vector called 'vertexSet' —> we will use this vector to create min heap containing vertices

inserting values in vectorSet

using in buld heap from standard template library <algorithm>

get the min element of the heap

visit that vertex

remove that vertex

same as above

loop for all adjacent vertices of the temp vertex

condition that vertex adjacent to temp and is in Q and w(u,temp) < key[v]

set values of parent

setting the key

heapify after removing the element

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

- prims.h
- prims.cpp

## 4.3 vertex Struct Reference

### Public Attributes

- int **key**
- int parent

  *the minimum value edge will be associated to its parent in minimum spanning tree(mst)*
- int value

*parent of the node in the mst*

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

- prims.cpp

# Chapter 5

# File Documentation

## 5.1 main.cpp File Reference

main execution file

```
#include <iostream>
#include "prims.cpp"
#include <string>
```

### Functions

- int **main** ()

### 5.1.1 Detailed Description

main execution file

## 5.2 prims.cpp File Reference

defintion file for prims.h

```
#include <cstdlib>
#include <iostream>
#include <algorithm>
#include "prims.h"
#include <vector>
```

### Classes

- struct vertex
- struct keyComparison

### 5.2.1 Detailed Description

defintion file for prims.h

**Author**

Nitesh Bhargava

**Date**

6/11/12

**approach:**

- here we have implemented the algorithm as described in the cormen book

## 5.3 prims.h File Reference

header file for the prims algorithm

### Classes

- class prims

### 5.3.1 Detailed Description

header file for the prims algorithm

**Author**

Nitesh Bhargava

**jos class:**

- public:
- init()

# Index

init
    prims, 7

keyComparison, 7
    operator(), 7

main.cpp, 11

operator()
    keyComparison, 7

prims, 7
    init, 7
prims.cpp, 11
prims.h, 12

vertex, 8