

# Data Structure Assignment 4

Course : data structure

Chapter : graph theroem

Implement language : C

Application : Metro route

## METRO of CITY A

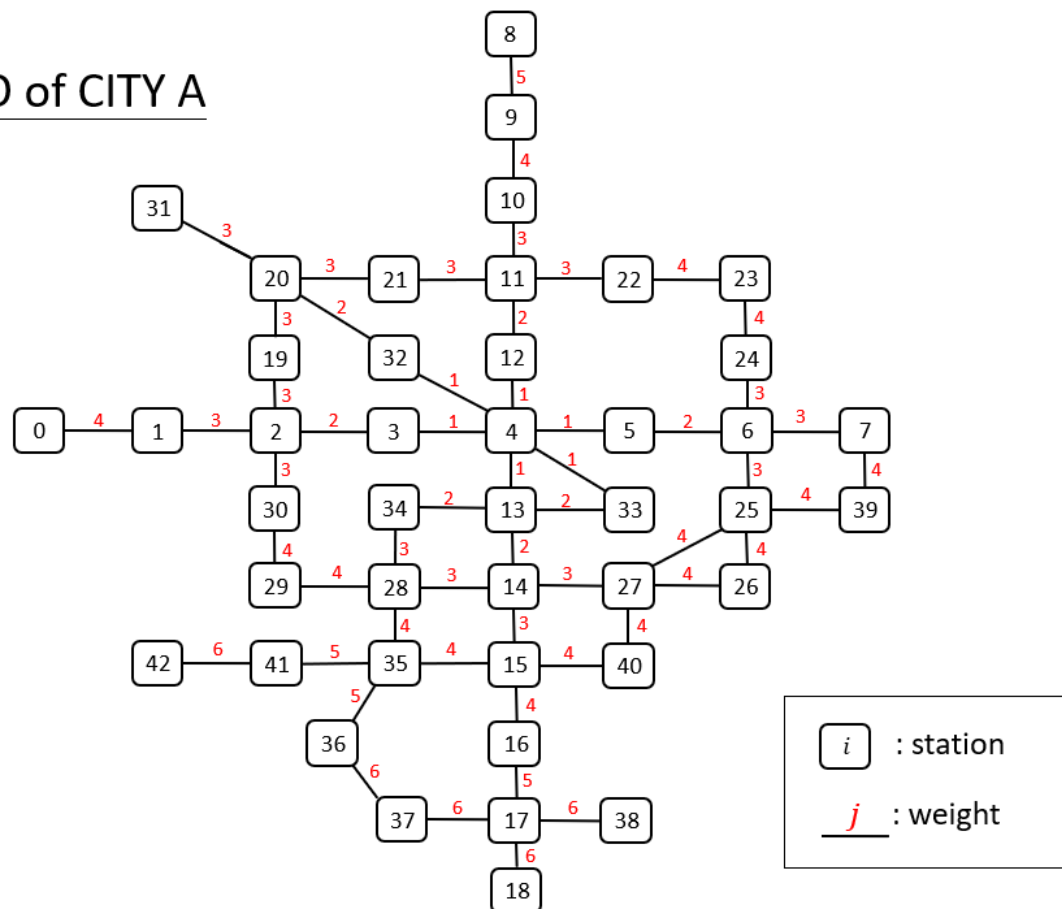


Figure 1

There are three questions below, please implement these questions by using C, and explain the process. The data structure for storing graph with less memory config may receive higher score.

Figure 1 shows the metro of City A, metro stores in the dataset file "example.txt", please read the dataset then construct it into a graph as adjacency matrix, adjacency list, ... and so on.

**Q1. Because the government has a limited budget for building metro at the early stage, please calculate the minimum cost for connecting all stations. We assume that the weight in the dataset (Fig.1) represents the cost of building the connection between two stations, e.g. a weight of 4 means the path costs \$4 million.(The weight of the graph is proportional to the distance between stations and station 4.)**

( Q1. solution\_1 ) : Using Prim's algorithm start from each node

```
===== QUESTION 1 RESULT (SOLUTION 1) =====
The minimum cost of building metro in the beginning is $137
=====
```

( Q1. solution\_2 ) : Using BFS start from node 4

```
===== QUESTION 1 RESULT (SOLUTION 2) =====
THE 1 LEVEL ::
  3   5  12  13  32  33
THE 2 LEVEL ::
  2   6  11  14  20  34
THE 3 LEVEL ::
  1   7  10  15  19  21  22  24  25  27  28  30  31
THE 4 LEVEL ::
  0   9  16  23  26  29  35  39  40
THE 5 LEVEL ::
  8  17  36  41
THE 6 LEVEL ::
 18  37  38  42

The minimum cost of building metro in the beginning is $137
=====
```

**Q2. Recently, the metro has completely constructed. Alan would like to visit the following stations : 29, 15, 2, 6, and start from station 36. Please help Alan to calculate the order of these stations with minimum number of stations that Alan will go through when he travel all these stations.**

( Q2. solution\_1 ) : Using greedy method (the answer is local optimization, i'll try other example)

```
===== QUESTION 2 RESULT (SOLUTION 1) =====
The order of visiting all node with minimum step ::
Start from node 4 2(with 2 step) 29(with 2 step) 15(with 3 step) 6(with 4 step)
The minimum total step is 11.
=====
```

( Q2. solution\_2 ) : Using Dynamic Programming

```
===== QUESTION 2 RESULT (SOLUTION 2) =====
The order of visiting all node with minimum step ::
PATH 1 : Start from node 4 2 29 15 6
PATH 2 : Start from node 4 6 2 29 15
The minimum total step is 11.
=====
```

**Q3. Now, Alan want to visit stations 2, 5, 26 and start from 37, Please compute the minimum cost between the three destinations and station 37 and show the paths that Alan would travel. Assume the weight in the dataset (Fig.1) represents the time it takes to travel between two stations. So, the answer should provide three different paths : 37 → 2, 37 → 5, and 37 → 26 with the minimum cost.**

( Q3. solution\_1 ) : Using Dijkstra's with stack and sorting

```
===== QUESTION 3 RESULT =====
Start from node 37 ::
Path of destination = 2 :
  37   17   16   15   14   13   4   3   2
  # Minimum cost of this path = 24
Path of destination = 5 :
  37   17   16   15   14   13   4   5
  # Minimum cost of this path = 22
Path of destination = 26 :
  37   17   16   15   14   27   26
  # Minimum cost of this path = 25
=====
```

( Q3. solution\_2 ) : Using Dijkstra's with minimum heap

```
===== QUESTION 3 RESULT =====
Start from node 37 ::
Path of destination = 2 :
  37   17   16   15   14   13   4   3   2
  # Minimum cost of this path = 24
Path of destination = 5 :
  37   17   16   15   14   13   4   5
  # Minimum cost of this path = 22
Path of destination = 26 :
  37   17   16   15   14   27   26
  # Minimum cost of this path = 25
=====
```

## Answer of the questions :

please check the code on git.

git : [https://github.com/jingyeyang/DataStructure\\_Graph.git](https://github.com/jingyeyang/DataStructure_Graph.git)

=====

檔案均於 "ds\_final\_graph" 中。

主要資料夾 :

```
ds_final_graph -- dataset (存放資料集)
                -- header (存放所有標頭檔)
                -- main.c (主要程式)
```

資料夾 "dataset" 底下檔案 :

```
ds_final_graph -- dataset -- example.txt (程式使用之資料集)
```

資料夾 "header" 底下檔案 :

```
ds_final_graph -- header -- dataStructure.h (存放使用之STRUCT)
                  -- preProcess.h (讀dataset並且建 Graph 成 adj list 跟 CSR
format)
                  -- stack.h (存放使用之 stack function)
                  -- queue.h (存放使用之 queue function)
                  -- minHeap.h (存放使用之 minimum heap function)
                  -- question1.h (存放第一題解答)
                  -- question2.h (存放第二題解答)
                  -- question3.h (存放第三題解答)
```

\*\*\*\*\* 所有解答於 question1.h, question2.h 和 question3.h 均有兩種版本, 以

```
#pragma region solution_1(2)
```

```
.....
```

```
.....
```

```
.....
```

```
#pragma endregion /* solution_1(2) */
```

做區分。\*\*\*\*\*

=====

執行方式 : ./run.sh 後產生 (bash 中路徑必須修改)

Compile Finish .....

start to Execute .....

ENTER DATASET NAME : 資料集名稱(example)