Data Structure Homework G : Graph

- *Objective*
  - Implement the Graph traversal algorithm : Breadth First Search
  - Find the Minimum Cost Spanning Tree using Kruskal's algorithm
  - Find the Shortest Path between two nodes using Dijkstra's algorithm
- *Description*
  - **Part one - Breadth First Search :**
    - Function details: your program should read the standard input and traverse the **DIRECTED** graph by Breadth First Search started from the start node in the command.
    - Test command : "executable file" –b "start name"
      - Sample command : hwd2 –b A
    - Input format :
      - First line : n
      - Other lines : "start name", "end name"
      - Range :

        n is an integer which represents the number of edges in a graph.

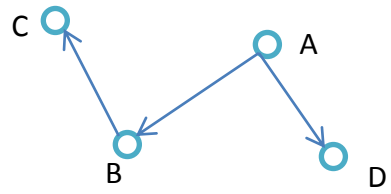
        A node name is a character in [A-Z]
      - Sample inputs :

        3

        A, B

        A, D

        B, C

    - Output format : "level order", "node name"
      - Sample outputs :

        0, A

        1, B

        1, D

        2, C

Part two - Minimum Cost Spanning Tree :

- Function details: your program should read the standard input and print out each edge of the minimum cost

spanning tree in the weighted undirected graph.

- Test command : “executable file” –m
  - Sample command : hwd2 –m
- Input format :
  - First line : n
  - Other lines : “start name”, “end name”, “weight”
  - Range :

    n is an integer which represents the number of edges in a graph.

    A node name is a character in [A-Z]

    A weight is an integer which represents the weight of an edge
  - Sample inputs :

    5

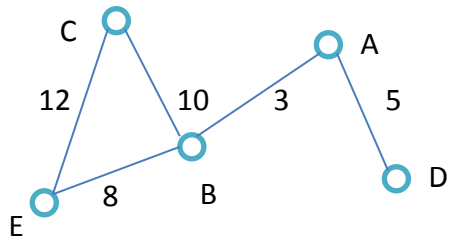
    A, B, 3

    A, D, 5

    B, C, 10

    C, E, 12

    B, E, 8



- Output format :
  - First line : “total cost”
  - Other lines : “start name”, “end name”, “weight”
  - Sample outputs :

    26

    A, B, 3

    A, D, 5

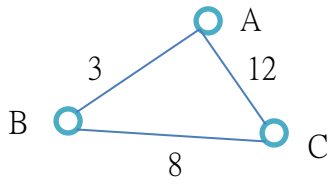
    B, E, 8

    B, C, 10

- **Part three - Shortest Path :**
  - Function details: your program should read the standard input and print out the shortest path between the two nodes in the weighted undirected graph.
  - Test command : “executable file” –s “start name” “end name”
    - Sample command : hwd2 –s A C
  - Input format :
    - First line : n

- Other lines : “start name”, “end name”, “weight”
- Sample inputs :

```
3
A, B, 3
A, C, 12
B, C, 8
```



- Output format :
  - First line : “total length”
  - Other lines : “node’s name”
  - Sample outputs :

```
11
A
B
C
```

- *Grade policies*
  - 5% - Source code can be compiled without any error
  - 20% - Part one : traverse graph with right order
  - 25% - Part Two : build the right MST using Kruskal’s algorithm
  - 30% - Part Three : find the right Shortest Path between two nodes using Dijkstra’s algorithm
  - 20% - readme file, code style, and comments in source code
- *Turn in*
  - System
    - Turn in files to the workstation : *csie0.cs.ccu.edu.tw*
    - Command: *turnin ds.hwG [files…]*
    - This source code will be compiled and tested on the workstation
  - Source code
    - Source code with appropriate comments
  - Report
    - A document named “readme.txt” or “readme.doc” or “readme.pdf”. you should describe the details of your project in your readme file