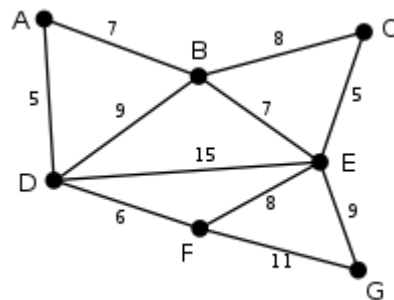


Exercise 4. Answer Sheet

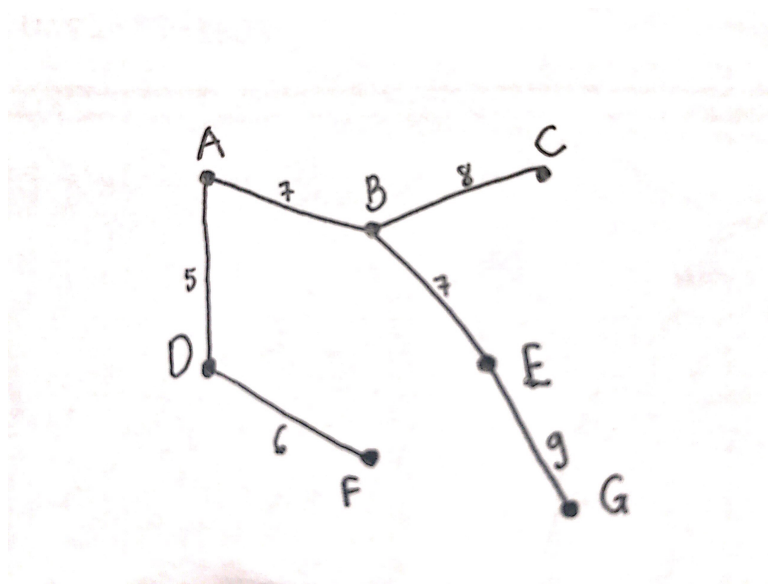
Student's Name: _____ Tran Thi Thoa _____

Student's ID: _____ s1242006 _____

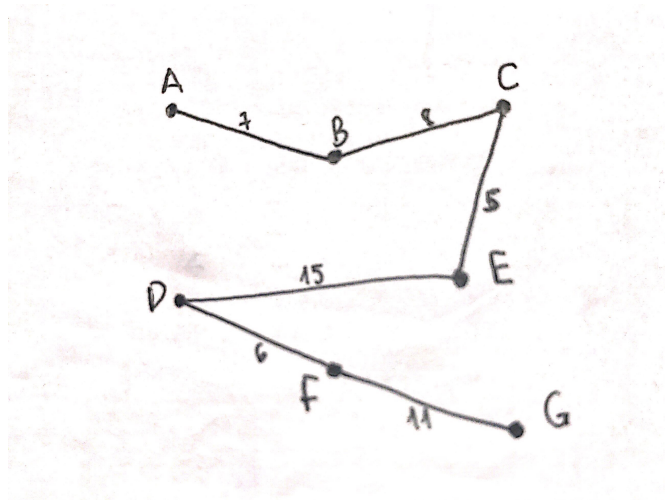
Problem 1. (50 points) Consider the following graph and assume node A as a root.



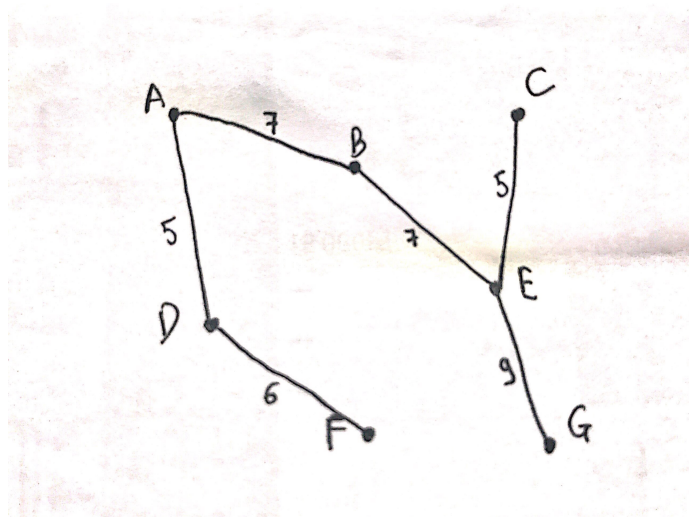
a) Draw a spanning tree obtained by using the Breadth First Search (BFS) algorithm.



b) Draw a spanning tree obtained by using the Depth First Search (DFS) algorithm.



c) Draw the minimum spanning tree obtained by the Prim's algorithm.



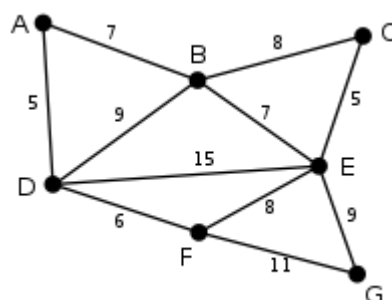
Problem 2. (50 points) Write a program implementing Kruskal's algorithm. Upload your source code. Show your input graph and the obtained MST in the space below.

The input graph is the same graph in the exercise 1

Because of the indexes in C++ programming language start from 0 therefore we consider the vertices of the graph : A is equal to 0, B is equal to 1, C is equal to 2, D is equal to 3, E is equal to 4, F is equal to 5, G is equal to 6.

The output will be the edges which will be added to the MST following the Kruskal algorithm and the total cost of the MST.

The graph we have :

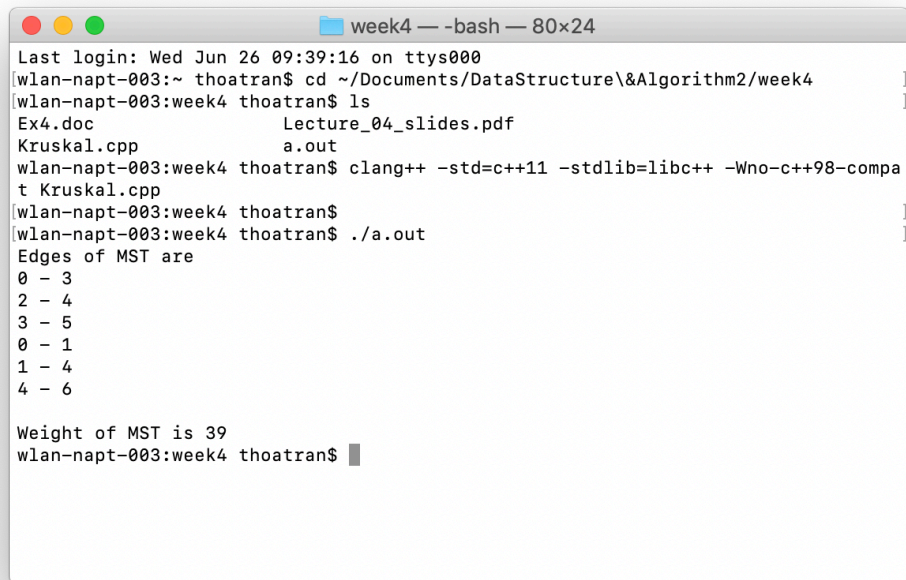


To implement and run the program, we change the directory to the directory we saved the Kruskal.cpp file and run the following command lines:

(Just for Mac OS)

```
clang++ -std=c++11 -stdlib=libc++ -Wno-c++98-compat Kruskal.cpp  
./a.out
```

For the above graph, the output will be:

A terminal window titled "week4 — -bash — 80x24" showing the execution of a C++ program. The user navigates to the directory ~/Documents/DataStructure/Algorithm2/week4 and lists the files, which include Ex4.doc, Lecture_04_slides.pdf, Kruskal.cpp, and a.out. The user then compiles Kruskal.cpp using clang++ with specific flags and runs the resulting a.out file. The output displays the edges of the Minimum Spanning Tree (MST) and its total weight.

```
week4 — -bash — 80x24  
Last login: Wed Jun 26 09:39:16 on ttys000  
wlan-napt-003:~ thoatran$ cd ~/Documents/DataStructure/Algorithm2/week4  
wlan-napt-003:week4 thoatran$ ls  
Ex4.doc          Lecture_04_slides.pdf  
Kruskal.cpp      a.out  
wlan-napt-003:week4 thoatran$ clang++ -std=c++11 -stdlib=libc++ -Wno-c++98-compat Kruskal.cpp  
wlan-napt-003:week4 thoatran$  
wlan-napt-003:week4 thoatran$ ./a.out  
Edges of MST are  
0 - 3  
2 - 4  
3 - 5  
0 - 1  
1 - 4  
4 - 6  
  
Weight of MST is 39  
wlan-napt-003:week4 thoatran$
```

We can see that the MST we have by using Kruskal algorithm is the same to the MST using Prim algorithm.