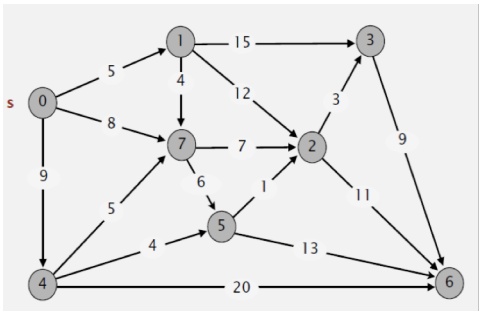
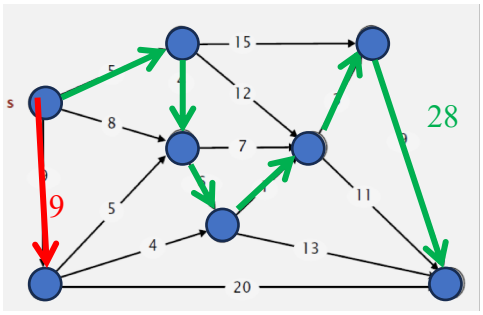
# Exercise 5. Answer Sheet

Student's Name: Haruki Terushima  Student's ID : s1290037

***Problem 1.*** *(15 points)* Consider the graph below.

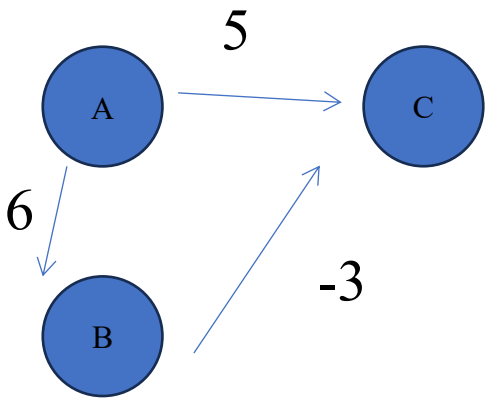



Draw a shortest path spanning tree with root at vertex ***s***. Show the cost (weight) of paths to each vertex.




**Problem 2**. *(15 points)* Dijksta's algorithm cannot handle negative weights. Show an example and explain what happens.

The negative weight changes the shortest path. There is an example;




To get C, there are 2 path: A=>B=>C and A=>C
If the weight of B to C is minus 3, then A, B, C path is the shortest.
But adding 3 for all the weight, then the shortest path changes to A to C.

***Problem 3.*** *(20 points)* Extend the pseudocode of the Bellman-Ford algorithm given at the lecture so it can detect negative cycles.

```
def Bellman-Ford-modified (G,s,w):
Init-SS (G,s)
for i=1 to |G.V|-1
    for each edge (u,v) in G.E
            RELAX (u,v,w)
           if d[u] + w(u, v) < d[v]:
                return "Negative cycle found"
```

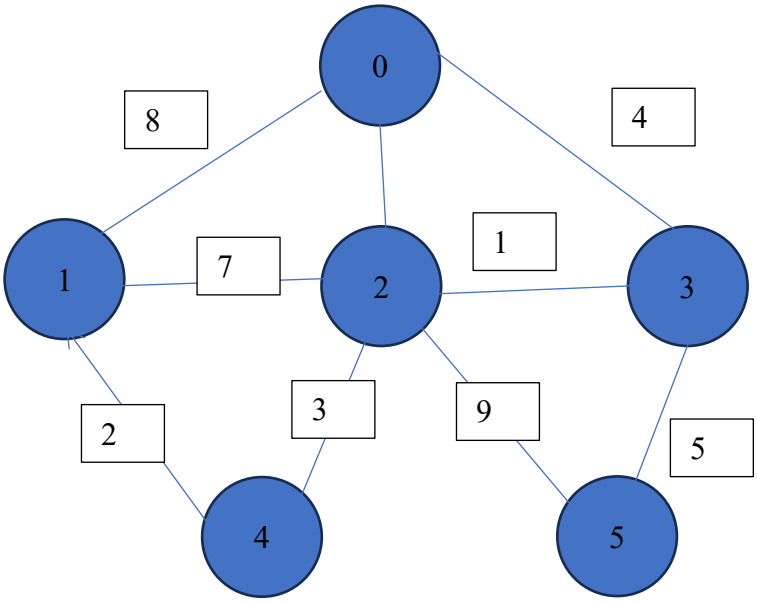
***Problem 4.*** *(50 points)* Write a program implementing Dijksta's algorithm. Upload your source code. Show your input graph and the obtained shortest path spanning tree in the space below.

***<Compire>***
***javac *.java***
***java findShortestPath***

***<input>***
***6***
***9***
***0 1 8***
***0 2 2***
***0 3 4***
***1 2 7***
***1 4 2***
***2 3 1***
***2 4 3***
***2 5 9***
***3 5 5***
***0***

***<output>***
***node 0 :***
***distance => 0 path =>***
***node 1 :***
***distance => 7 path => 0 2 4***
***node 2 :***
***distance => 2 path => 0***
***node 3 :***
***distance => 3 path => 0 2***
***node 4 :***
***distance => 5 path => 0 2***
***node 5 :***
***distance => 8 path => 0 2 3***

< input graph >




< shortest path spanning tree >