Prim's vs Dijktra's Algorithms and its Analysis

Dania Farooq
Sargodha Department of the computer science University of Lahore 19sb tehsil kotmomin district Sargodha in Pakistan

Abstract

Where vary the difference between prims and dijktra algorithms which both relay on same concept but solve two different problems. Prims is the minimum spanning tree and used for a graph and provide shortest path. Dijktra are also use for shortest path and to find out the shortest path, so they are little differ with each other. In our research paper we will display the prims and dijktra algorithms with the help of example to solve out the problems and will show which the best to use to solve the problems is.

I. INTRODUCTION

Prim's algorithm and Dijkstra's algorithm have the same idea but these are solve in two different problems. Prim's algorithm finds a minimum spanning tree for a graph and adding the shortest edge to connect the node while the Dijkstra's algorithm ideas relay on shortest path tree starting from some source node. A shortest path tree is a tree that connects all nodes in the graph and it show the minimum graph of node it provide shortest path Of node but it can be cost could be much larger than the cost of an MST, because the shortest path tree is not guaranteed to be a minimum spanning tree, so its cost can be larger.The prims are undirected graphs but Dijkstra's are directed graph and cannot handle the negative edge weights of the graphs prims can handle all the problems and can be solve the negative edge weights in the graphs but dijktra's cannot handle so it's best to choose the prims algorithm

II. PRIM'S ALGORITHM

A. Definition

Prims is the minimum spanning tree which provide the shortest path .it's the undirected graph and can handle the negative the edge weights in the graph it work with the same like the dijktra algorithm but solve the different problems .the diagram are shown in figure 1:

B. Solution

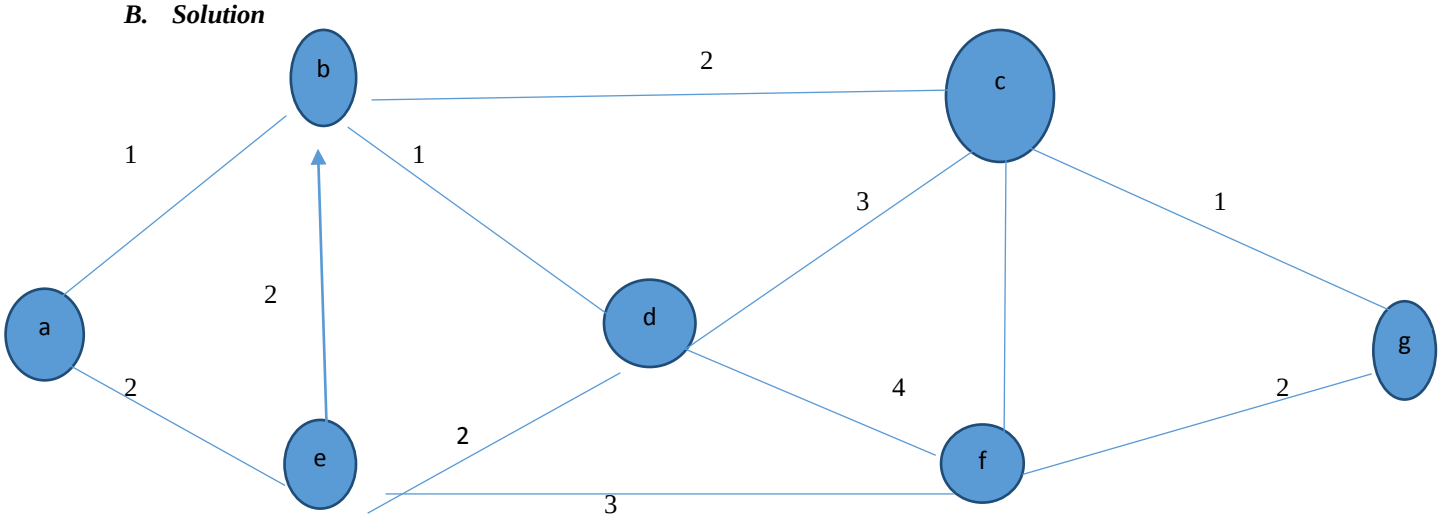

Figure 1: Prim's graph

We now solve this in this method:

In this we first make the key table where we insert all the nodes in the line from node A to node G. after that we place the ∞ in every under the node .first we start from the node A, a node has the 0 value .we place 0 value into the ∞ because 0 is less than infinite .when we start from the b we add the value of edge of the b like we add only 1 that is the value of the edge of the b we don't add previous value of the node. We don't add previous value in prims we just add current value of the edge in prims algorithm .when the value will be less than other value or the infinite we place small value rather than greater value. so when we go to b node we place 1 value in the infinite , when we start from a we check where is a node A edge are going then we place value in that node like a edge are going to B and E node we add edge value and remove infinite

.after that we cut a and move to next to check which node have less value and we select that node to proceed next .after A b have less value and we select b node and we check where is b node going and we than replace less value in every node where the infinite are exit .we do all process same for every node if the node have greater value we replace with less value .when we reached at node f the node f have greater value like 4 ,when node e goes to f it have 3 value we replace 4 by 3 and next we fined 2 that is less than 3 so we again replace 3 by 2 and that is final value .

All these process and its value are show in KEY table like this:

C. Key Table

a	b	c	d	e	f	g
~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
0	1	2	1	~~2~~	~~4~~	1
				1	~~3~~	
					2	

Table 1: Prim's key

We again made a table that show also KEY and PARENT ,parent are the node that node are change to node when we change the previous value like 2 are replace it by 1 in E node and in F node 4 replace to 3 and 3 replace to 2 in this that not only value replace also it parent change like if node E has A node after change its value its parent or node are also change and E node will have node B its node will change ,F node have first time D node after it have E but again its value change it have node C and its final parent are now C node this process are show in table First time we will assign every parent as a NILL node after when the node change its parent NILL will be removed

D. Parent Table

NODES	KEY	PARENT
A	~~∞~~ 0	~~N~~
B	~~∞~~ 1	~~N~~ a
C	~~∞~~ 2	~~N~~ b
D	~~∞~~ 1	~~N~~ b
E	~~∞~~ ~~2~~ 1	~~N~~ ~~a~~ b
F	~~∞~~ ~~4~~ ~~3~~ 2	~~N~~ ~~d~~ ~~e~~ c
G	~~∞~~ 1	~~N~~ c

Table 2: Prim's Parent

E. Shortest Graph

We will finally make graph according to parent node this is the final graph of the PRIMS algorithm. In this all node are connected with each other and can reached one node to another nod

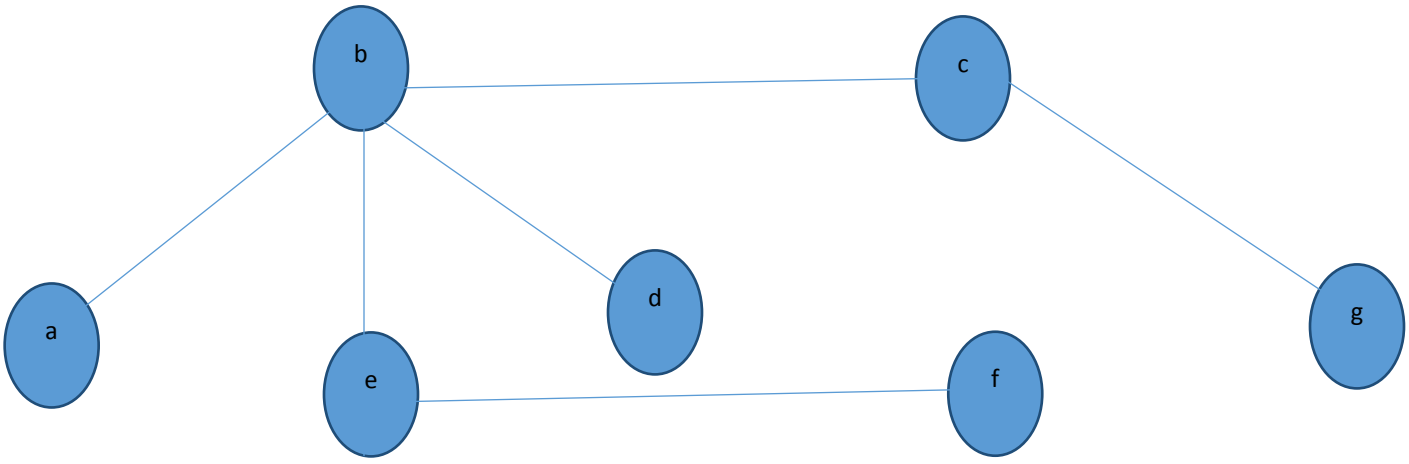

Figure 2:Prim's graph

F. Algorithm

```
Prim (G, w, r) {
   for each (u in V)
{
 Key[u] = ∞
     Color[u] = white;
}
   Key[r] = 0;
   pred[r] = nil;
   Q = new PriQueue(V);
   While (Q. 0 ())
{
      u = Q.extractMin();
      for each (v in adj[u])
{
if ((color[v] == white) &
            (w(u,v) < key[v])
        key[v] = w(u, v);
{
   }
}
      color[u] = black;
}
  }
```

G. Time Complexity of Prims Algorithm

O(log n)

III. DIJKTRA'S ALGORITHM

A. Definition

Dijktra algorithm is consider as the shortest path algorithm .it is directed algorithm and can't handle the negative edge weight of the graph.

B. Solution

We will resolve dijktra's algorithm by using directed graph its idea same about the prims algorithm but it works different and use for directed graphs.

C. Graphs

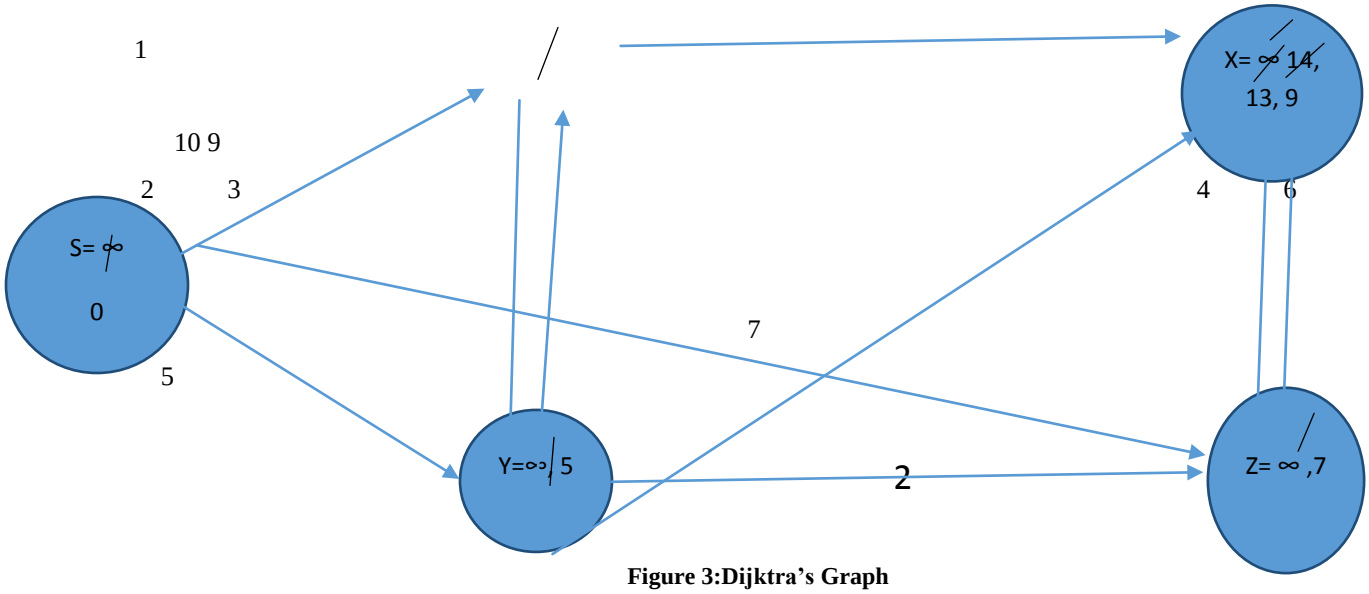

Figure 3:Dijktra's Graph

D. Directed Graph for Dijktra's Algorithm

Dijktra's is the directed graph which is used for the finding shortest path this is the same idea like prims algorithm but in this algorithm we add precious value of the edges of the node. First we start from the node S which have 0 value and its edges are going to node T and node Y its edges are directed .first time the value of the node is assign infinite after that we add the value of the edges and the previous value of the node we add the value 0 of the node S into node T edge like 0 +10=10 and 10 < ∞ we add this value into the node T and remove infinite and add the value into the table now T node parents will b node S that will also insert into the parent table. In the same way we add 0 into the edge of y like 0=+5=5 and add the value into the KEY table and its parent will be add into the parent table its parent will also node S. after that check these value we find the small value in between nodes and then we select less value of the node ,again we check where that node's edges going and then we add previous +current value and add that value into the Key table and its parent also add into the Parent table.in X node first we add 14 then 13 and then 9 because when we add "previous +current" value the next new value are less than previous value so we remove previous value and add new and less value into the table and due to change the value we also change the parent and update the parent table according to its node and parent .These are some steps to follow and solve the problem using dijktra's algorithm in the Figure 3in this way we find the shortest graphs and shortest path.

E. Key Table

S	T	X	Y	Z
∞	∞	∞	∞	∞
0	10	14	5	7
	8	13		
		9		

Table 3: Dijktra's Key

F. Parent Table

NODES	KEY	PARENT
S	∞ 0	N
T	∞ 10, 8	N S Y
X	∞ 14, 13, 9	N Y Z T
Y	∞ 5	N S
Z	∞ 7	N Y

Table 4: Dijktra's Parent

After this we will draw the shortest graph that show the shortest path and shown in figure 4

G. Final Graph

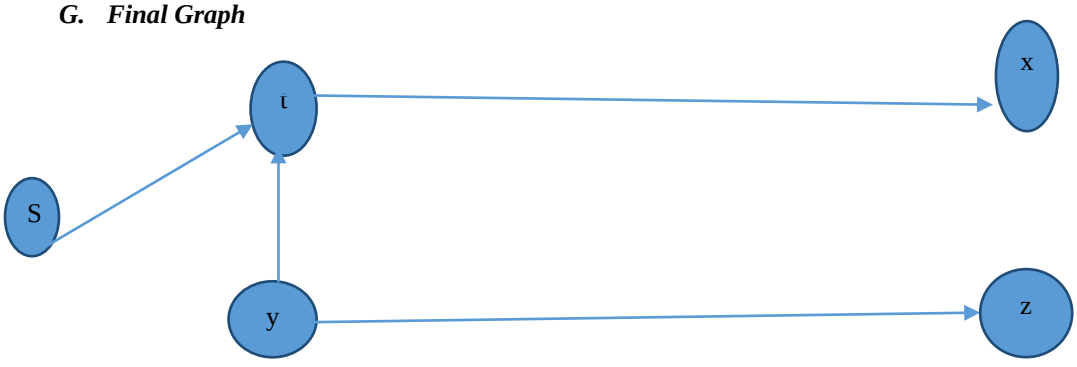

Figure 4: Dijktra's Graph

H. Dijktra Algorithm

```
Dijktra (G,W, S)
 INITILIZE SINGLE SOURCE (G,w,S)
        S=0
        Q=G.V
        While Q ≠ 0
                U.EXTRACT_MIN(Q)
                S= SU{U}
                For each vertex vƐG. adj[u]
                        RELAX (u, v, w).
 INITILIZE SINGLE SOURCE (G,S)
        For each vertex vƐG.v
                v.d = nil
                s.d = 0
RELAX (U,V,W)
        If v.d>u.d + w(u,v)

                v.d = u.d + w(u,v)
                        v.π = u.
```

I. Time Complexity of Dijktra's Algorithm

In worst cane O(|E|+|V|log|V|)

IV. DIFFERENCE BETWEEN PRIM'S AND DIJKTRA'S ALGORITHM

A. Prim's Algorithm

1. It's the minimum spanning tree
2. Undirected graphs
3. It can solve the negative edges weight of the graphs
4. Best for use as compare to dijktra because it handle the negative weight edges of the graphs
5. It add only current value or current weights of the edges to the node
6. Prim's algorithm returns graph as argument, and take as a tree.
7. It has different signature as compare to dijktra's
8. Prims as a greedy choice it choose the edge of minimum weight that crosses the (S-V).
9. It is consider as a simple in greedy choice.

B. Dijktra's Algorithm

1. It's the shortest path
2. directed graphs
3. it cannot handle the negative edges weight of the graphs
4. it add previous value + current value of the edges of the node
5. it is less efficient as compare to Prim's algorithm because it cannot handle the negative weights
6. Dijkstra's algorithm returns the graph and the starting node as arguments.
7. It also takings a function that gives shortest paths for each node.
8. It also has different signature as compare to Prim's algorithm
9. Dijktra's chooses the minimum distance from the source of vertex s, in greedy choice
10. It consider as a more complicated in greedy choice.

V. CONCLUSION

In this paper we know that how to solve the graph by using two different algorithm to find the shortest path .in this we know that the use of prims algorithm is the best way to find the shortest path because it can solve negative weights but as compare to dijktra'salgorithm it have no grip on negative weights, so we got that Prim's are best to use to catch the shortest path. Dijktra's use short path. it can have grip on negative and non-negative path it can solve cyclic or acyclic problem.

VI. FUTURE WORK

As concern in dijktra's algorithm to solve the shortest path must be permit the distance should be negative not only academic in nature. To find out the shortest path we can use all graphs but all graphs have different way to find out the path prims handle negative weights but dijktra not,so the Prims are the best to use the Prims algorithm as parallel to dijktra's algorithm .in dijktra's it's not necessary it cannot handle negative weights we can make dijktra's also handle negative weights due to change its code little bit are can change it to undirected graphs to use another techniques etc. .It need a lot more work in future so that it will be capable to handle the negative weights.

REFERENCE

[1] "Prims algorithm introduction"/Wikipedia/wiki/
[2] "Dijktra's algorithm introduction"/ html/wiki/
[3] Difference between Prims and dijktra's algorithm /wiki/ notes/
[4] Quora "Difference b/w Prims and dijktra's /
[5] www.ms.unimelb.edu/"Dijktra"/mosh
[6] "Prims algorithm"/codes/notes/
[7] "Prims complexity"/ 27/notes/Lo7/207/
[8] "Dijktra's detail"/wiki/org.notes/
[9] "Dijktra's introduction"/video/institute/
[10] www.Prims.org/ introduction/notes/codes