**PRIMS ALGORITHM:**

Prim's algorithm is a greedy algorithm. It treats node as a single tree and keeps adding node to the spanning tree. We start with one node and then keep adding the edges that have minimum weight until all nodes are visited.

Here are steps we use to perform the Algorithm

1. We ask user to select the Prims algorithm.
2. Then we ask user to enter the number of nodes.
3. We call function prims ().
4. We note the start time at the start of function.
5. We create the randomize matrix of nodes\*nodes
6. We create three arrays visited, parent, current.
7. Current keeps the track of current key of vertices, Parent keeps the track of parent of vertices and visited keeps the track of all the visited nodes.
8. Find the key with minimum value and not yet visited
9. Find all the adjacent node and update the weight of the vertices
10. Repeat till all the nodes are visited

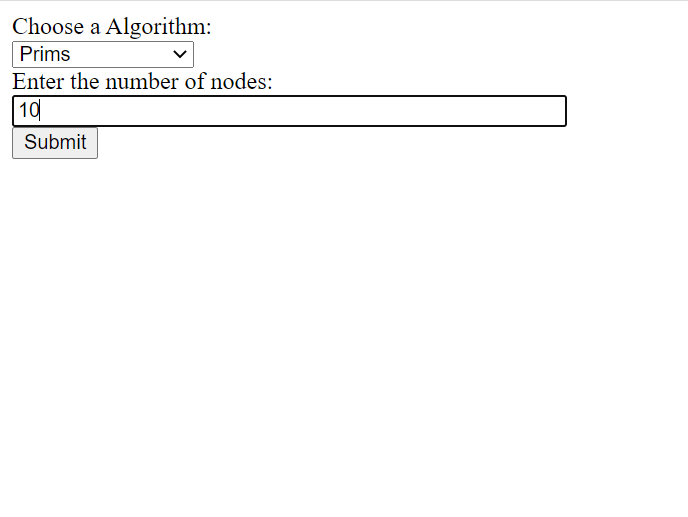
**Kruskal’s Algorithm:**

Kruskal algorithm is greedy algorithm. It picks the smallest edge in the graph that does not cause cycle.

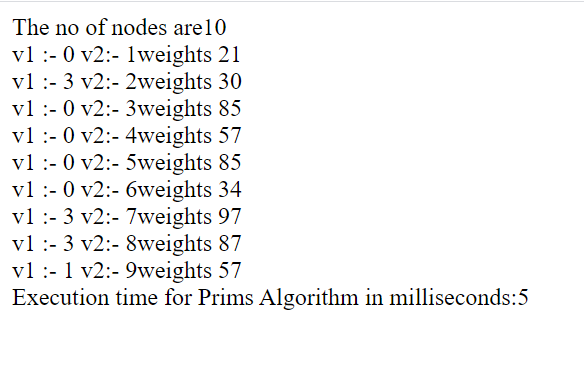
Here are steps we use to perform the Algorithm

1. We ask user to select the Kruskal algorithm.
2. Then we ask user to enter the number of nodes.
3. We call function kruskal ().
4. We note the start time at the start of function.
5. We create the randomize matrix of nodes\*nodes and weight.
6. We sort the weight in ascending order.
7. We keep the track of parent and rank.
8. We keep moving forward and keep checking that no cycles are created

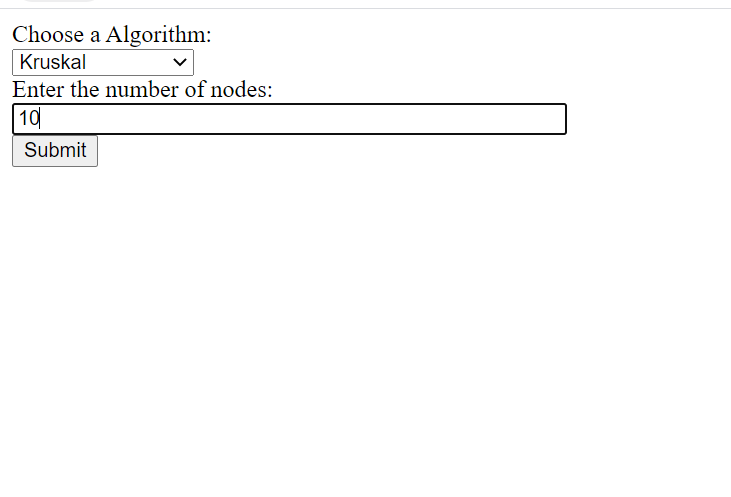
**GUI Design Screen Shot:**



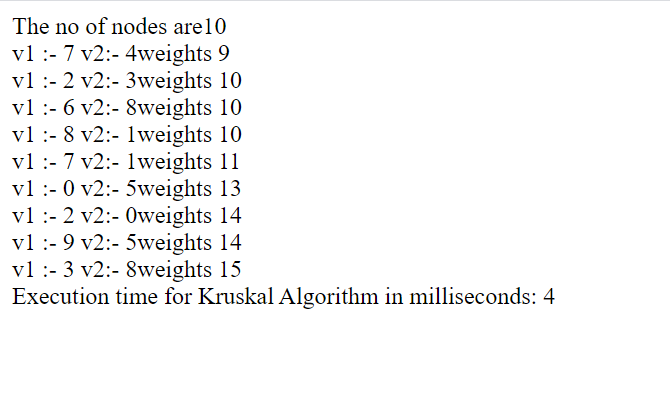
**Result Screen:**



**Kruskal Algorithm:**



**Result Screen:**



**Time Comparison for 10 nodes**

|  |  |
| --- | --- |
| **Algorithm** | **Time(milliseconds)** |
| **Prims** | **5** |
| **Kruskal** | **4** |

**Run time comparison between Prims and Kruskal:**