Graphs

LAB-7

Priyansh Bakhtani

PRN - 20190802079

1.)Aim: To compute MST using kruskal's algorithm and prim's algorithm and to fid the shortest path using dikstra's algorithm.

2.)Theory:

1.)Minimum spanning tree(MST): A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible.

2.)Kruskal's Algorithm: Kruskal's algorithm to find the minimum cost spanning tree uses the greedy approach. This algorithm treats the graph as a forest and every node it has as an individual tree. A tree connects to another only and only if, it has the least cost among all available options and does not violate MST properties.

- **Algorithm:**

  1.Sort all the edges in non-decreasing order of their weight.
  2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.
  3. Repeat step#2 until there are (V-1) edges in the spanning tree.

3.)Prim's Algorithm: Prim's Algorithm is used to find the minimum spanning tree from a graph. Prim's algorithm finds the subset of edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized. Prim's algorithm starts with the single node and explore all the adjacent nodes with all the connecting edges at every step. The edges with the minimal weights causing no cycles in the graph got selected.

- **Algorithm:**

1) Create a set min_span_tree_set that keeps track of vertices already included in MST.

2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE.

3)Assign key value as 0 for the first vertex so that it is picked first.

4)While min_span_tree_set doesn't include all vertices

a) Pick a vertex u which is not there in min_span_tree_set and has minimum key value.
b) Include u to min_span_tree_set.
c)Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if weight of edge u-v is less than the previous key value of v, update the key value as weight of u-v The idea of using key values is to pick the minimum weight edge from cut. The key values are used only for vertices which are not yet included in MST, the key value for these vertices indicate the minimum weight edges connecting them to the set of vertices included in MST.

4.)Dijkstra's Algorithm: Dijkstra's algorithm is very similar to Prim's algorithm for minimum spanning tree. Like Prim's MST, we generate a *SPT (shortest path tree)* with given source as root. We maintain two sets, one set contains vertices included in shortest path tree, other set includes vertices not yet included in shortest path tree. At every step of the algorithm, we find a vertex which is in the other set (set of not yet included) and has a minimum distance from the source.

- **Algorithm:**

1) Create a set *sptSet* (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

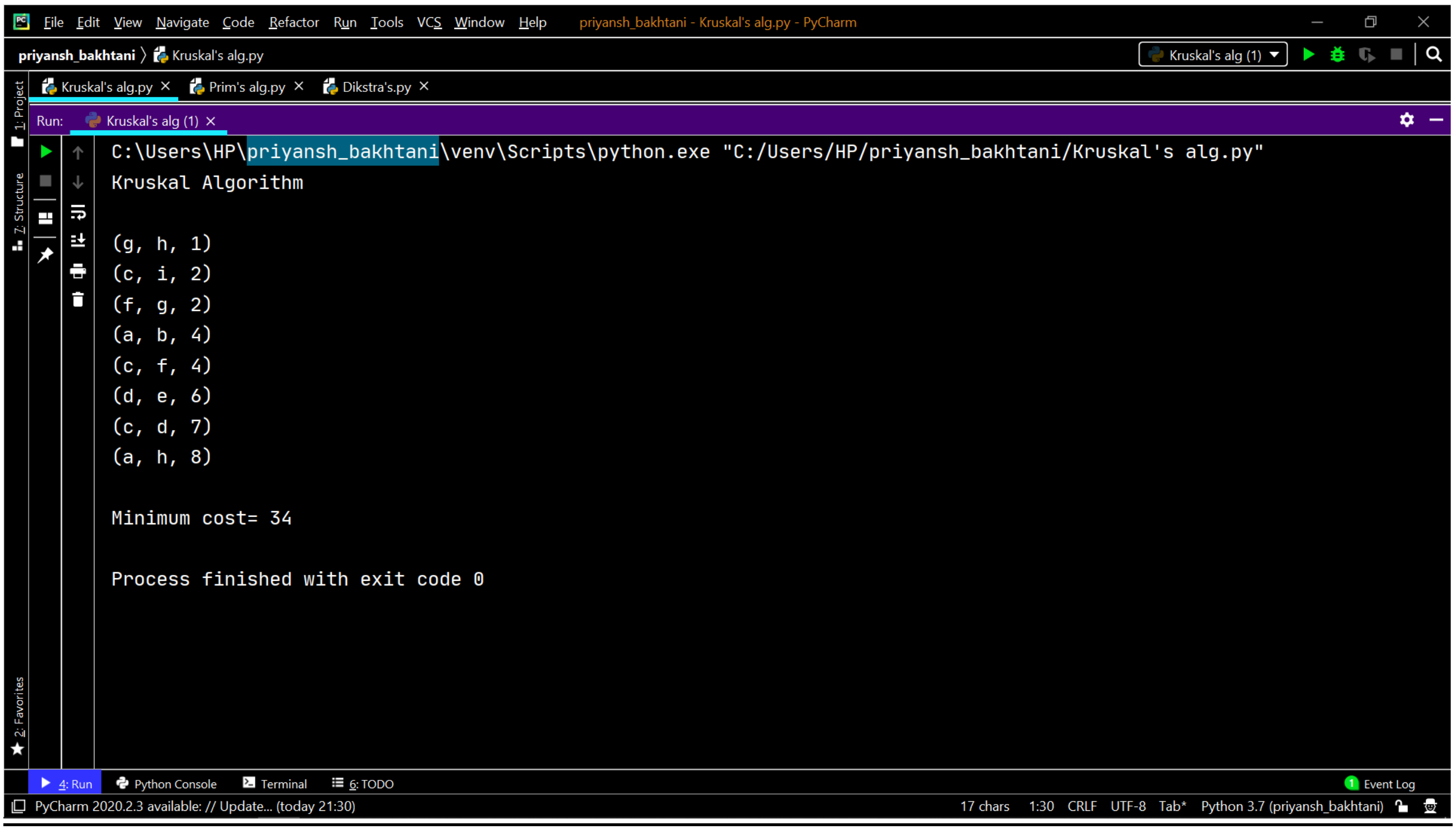
2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

3) While *sptSet* doesn't include all vertices

....**a)** Pick a vertex u which is not there in *sptSet* and has minimum distance value.
....**b)** Include u to *sptSet*.
....**c)** Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.

Outputs:

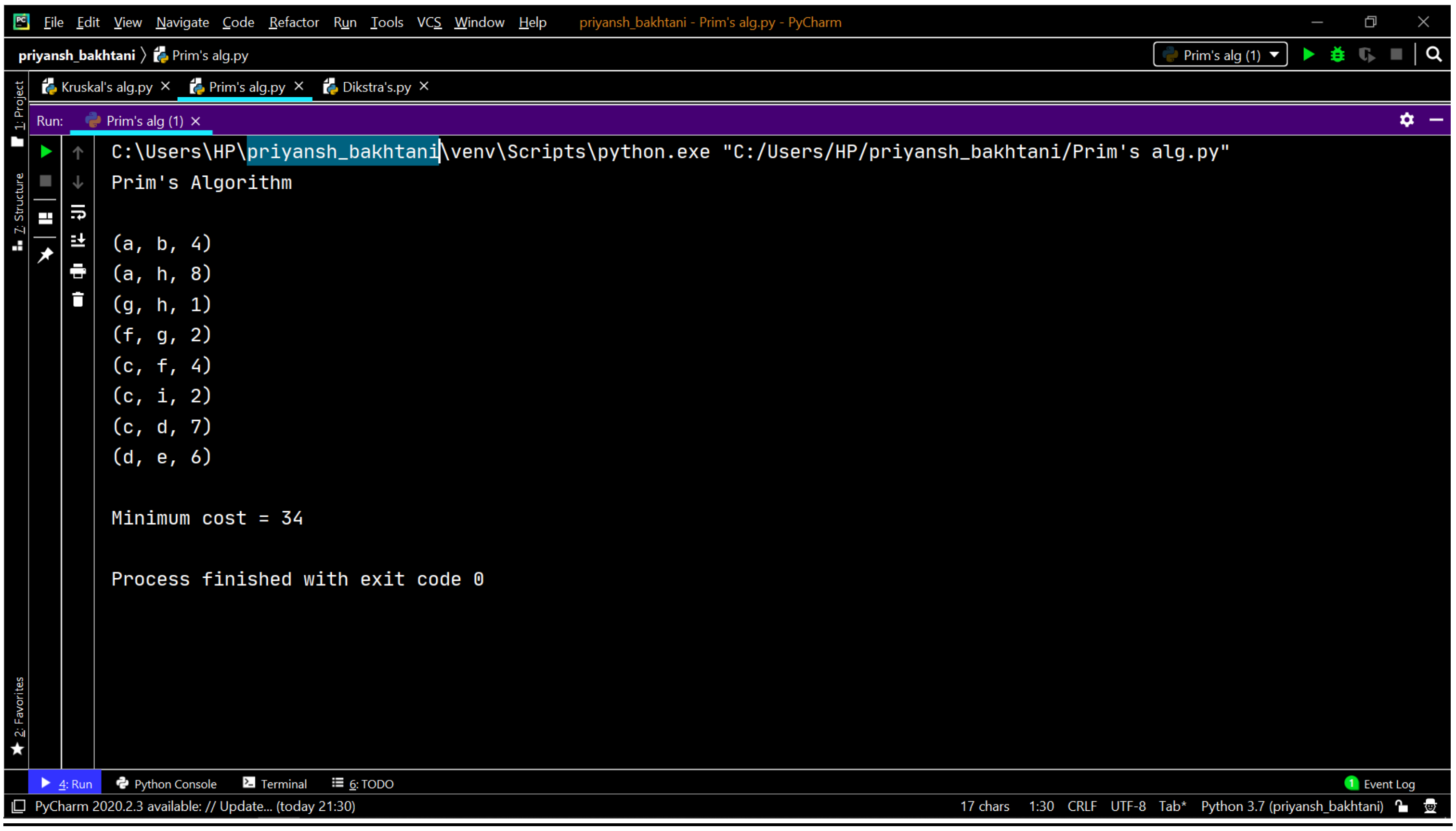
1.)Kruskal's Algorithm:



```
C:\Users\HP\priyansh_bakhtani\venv\Scripts\python.exe "C:/Users/HP/priyansh_bakhtani/Kruskal's alg.py"
Kruskal Algorithm

(g, h, 1)
(c, i, 2)
(f, g, 2)
(a, b, 4)
(c, f, 4)
(d, e, 6)
(c, d, 7)
(a, h, 8)

Minimum cost= 34

Process finished with exit code 0
```

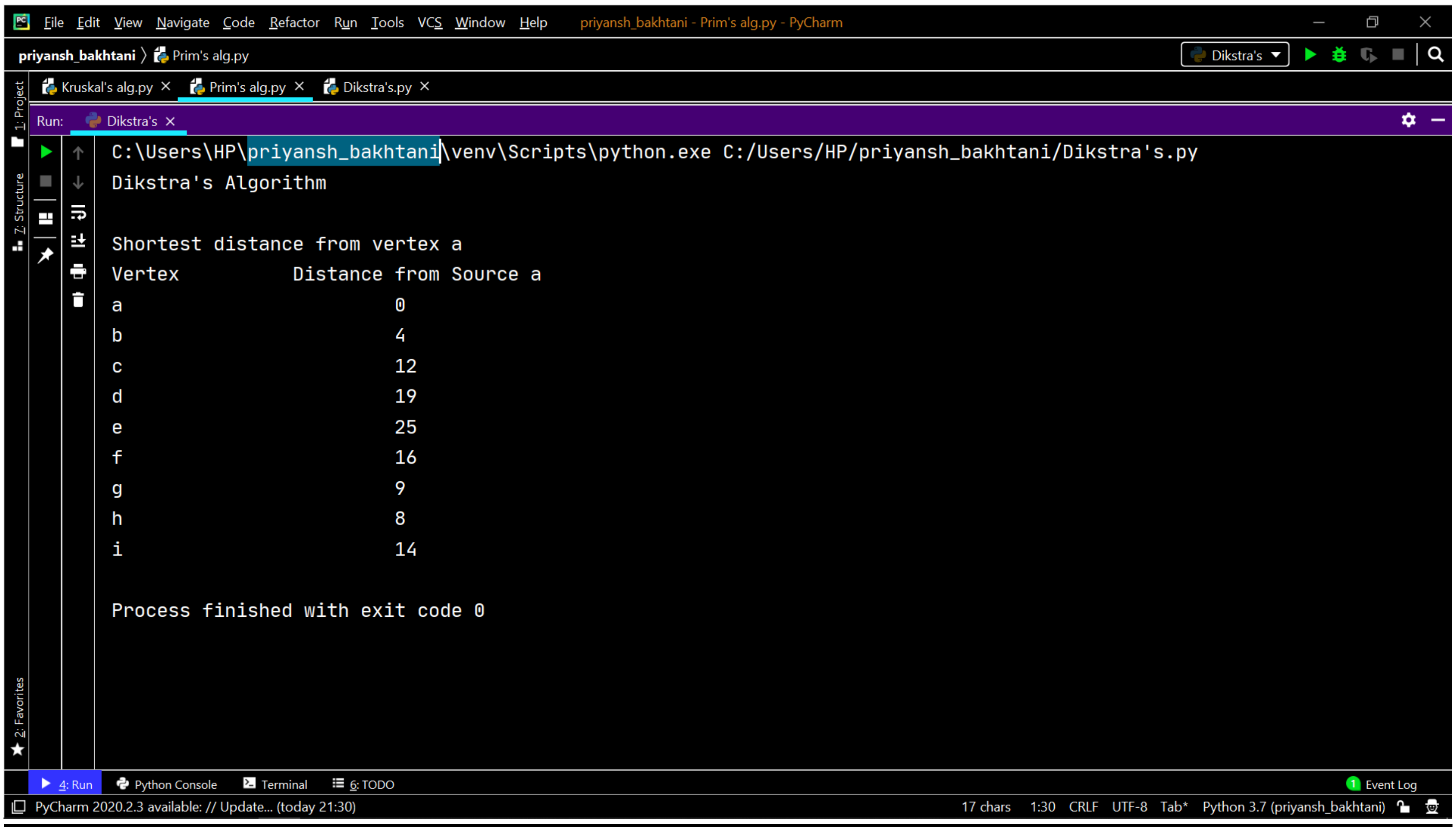
File Edit View Navigate Code Refactor Run Tools VCS Window Help
priyansh_bakhtani - Kruskal's alg.py - PyCharm
priyansh_bakhtani 〉 Kruskal's alg.py
Kruskal's alg (1)
Kruskal's alg.py
Prim's alg.py
Dikstra's.py
Run: Kruskal's alg (1)
1: Project
7: Structure
2: Favorites
4: Run
Python Console
Terminal
6: TODO
Event Log
PyCharm 2020.2.3 available: // Update... (today 21:30)
17 chars 1:30 CRLF UTF-8 Tab* Python 3.7 (priyansh_bakhtani)

2.)Prim's Algorithm:


File Edit View Navigate Code Refactor Run Tools VCS Window Help priyansh_bakhtani - Prim's alg.py - PyCharm
priyansh_bakhtani 〉 Prim's alg.py
Prim's alg (1)
Kruskal's alg.py × Prim's alg.py × Dikstra's.py ×
Run: Prim's alg (1) ×
4: Run Python Console Terminal 6: TODO Event Log
PyCharm 2020.2.3 available: // Update... (today 21:30) 17 chars 1:30 CRLF UTF-8 Tab* Python 3.7 (priyansh_bakhtani)

```
C:\Users\HP\priyansh_bakhtani\venv\Scripts\python.exe "C:/Users/HP/priyansh_bakhtani/Prim's alg.py"
Prim's Algorithm

(a, b, 4)
(a, h, 8)
(g, h, 1)
(f, g, 2)
(c, f, 4)
(c, i, 2)
(c, d, 7)
(d, e, 6)

Minimum cost = 34

Process finished with exit code 0
```

3.)Dikstra's Algorithm:


File Edit View Navigate Code Refactor Run Tools VCS Window Help priyansh_bakhtani - Prim's alg.py - PyCharm
priyansh_bakhtani 〉 Prim's alg.py
Dikstra's
Kruskal's alg.py × Prim's alg.py × Dikstra's.py ×
Run: Dikstra's ×
4: Run Python Console Terminal 6: TODO Event Log
PyCharm 2020.2.3 available: // Update... (today 21:30) 17 chars 1:30 CRLF UTF-8 Tab* Python 3.7 (priyansh_bakhtani)

```
C:\Users\HP\priyansh_bakhtani\venv\Scripts\python.exe C:/Users/HP/priyansh_bakhtani/Dikstra's.py
Dikstra's Algorithm

Shortest distance from vertex a
Vertex          Distance from Source a
a                        0
b                        4
c                        12
d                        19
e                        25
f                        16
g                        9
h                        8
i                        14

Process finished with exit code 0
```

- **Conclusion:** The MST using kruskal's and prim's method has been calculated .Also shortest distance using dikstra's algorithm has been calculated.