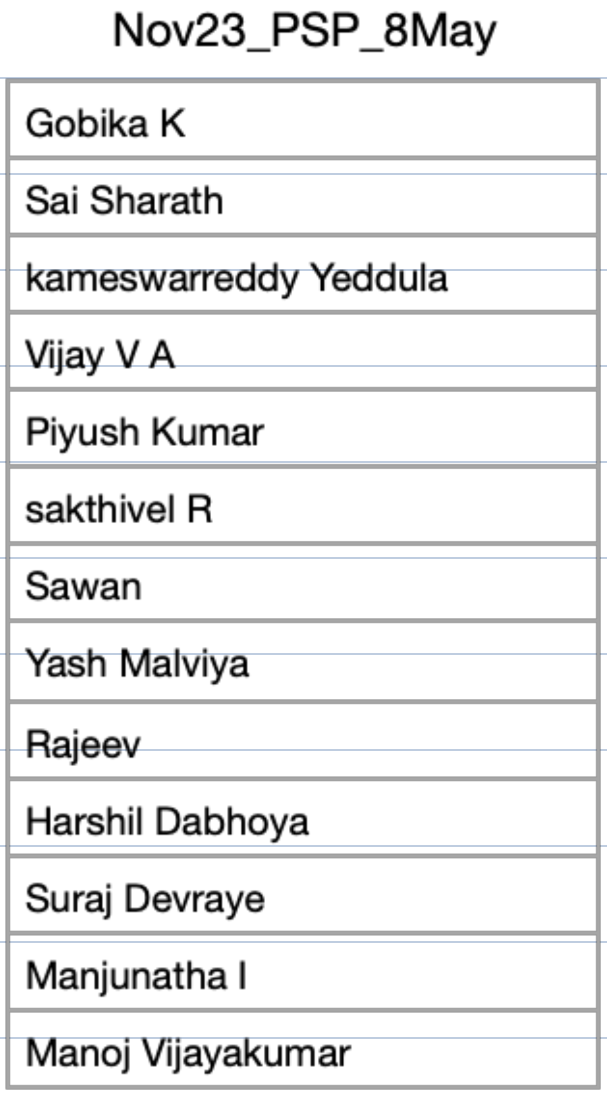
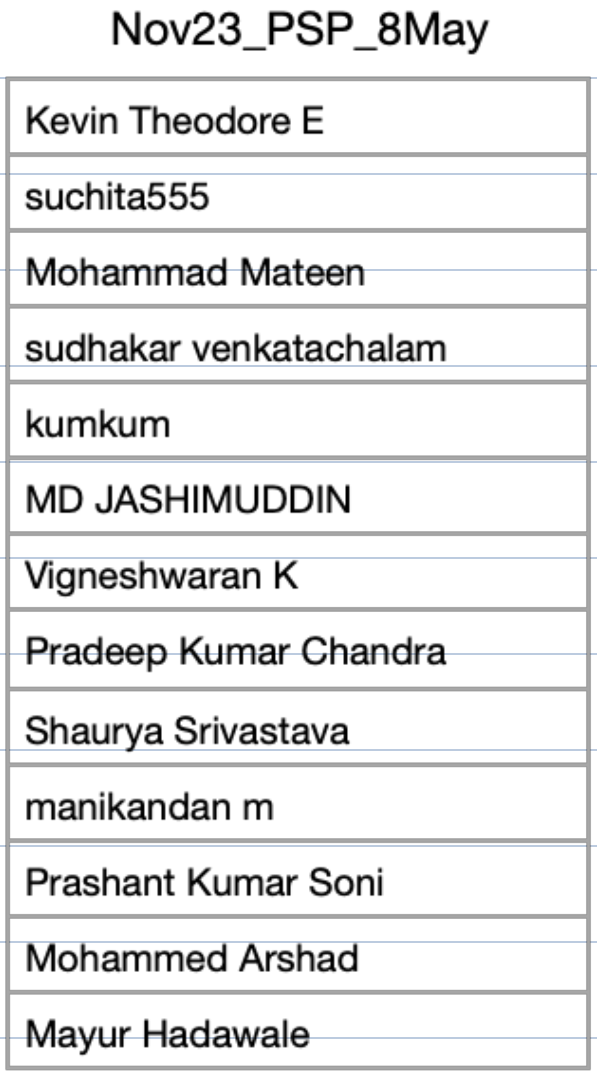
Current PSP 52.04 → 60 %



Nov23_PSP_8May
Gobika K
Sai Sharath
kameswarreddy Yeddula
Vijay V A
Piyush Kumar
sakthivel R
Sawan
Yash Malviya
Rajeev
Harshil Dabhoya
Suraj Devraye
Manjunatha I
Manoj Vijayakumar



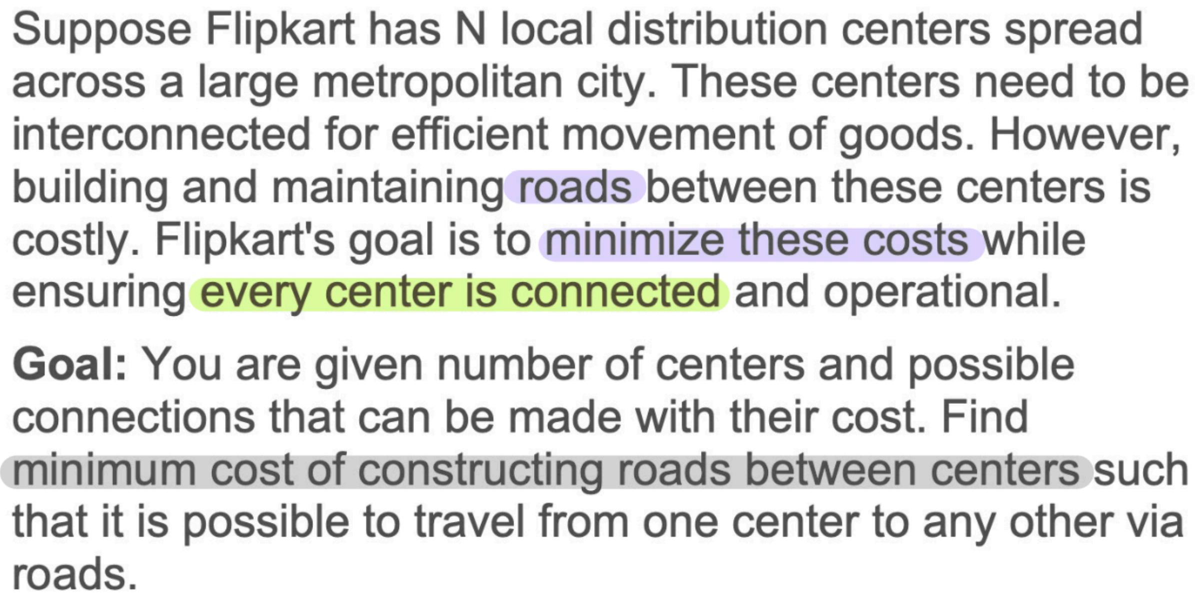
Nov23_PSP_8May
Kevin Theodore E
suchita555
Mohammad Mateen
sudhakar venkatachalam
kumkum
MD JASHIMUDDIN
Vigneshwaran K
Pradeep Kumar Chandra
Shaurya Srivastava
manikandan m
Prashant Kumar Soni
Mohammed Arshad
Mayur Hadawale

Agenda

a) Flipkart case study

b) Prims Algorithm

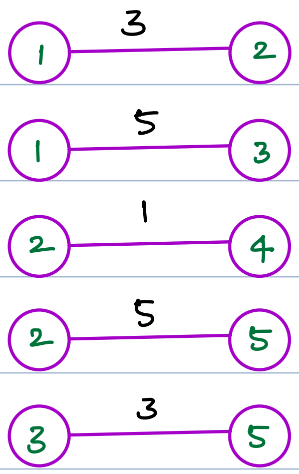
c) Dijkstra's Algorithm

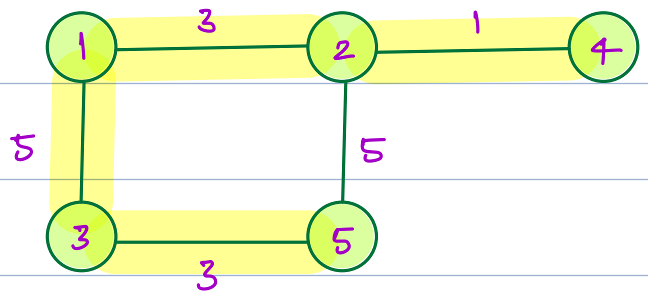


Suppose Flipkart has N local distribution centers spread across a large metropolitan city. These centers need to be interconnected for efficient movement of goods. However, building and maintaining roads between these centers is costly. Flipkart's goal is to minimize these costs while ensuring every center is connected and operational.

Goal: You are given number of centers and possible connections that can be made with their cost. Find minimum cost of constructing roads between centers such that it is possible to travel from one center to any other via roads.

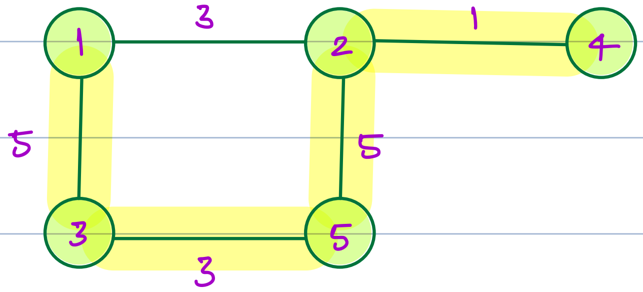
N = 5





MST

If a graph is connected and has N-1 edges → tree



not MST

Minimum Spanning Tree {MST}

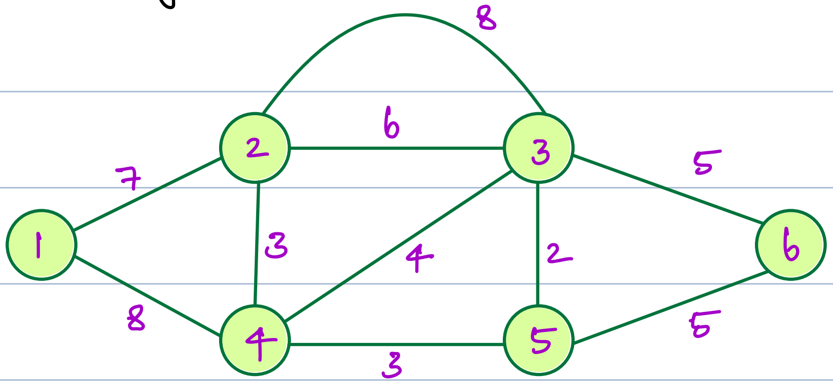
Tree generated from a connected graph, such that

- all nodes are connected
- sum of weights of all selected edges is min

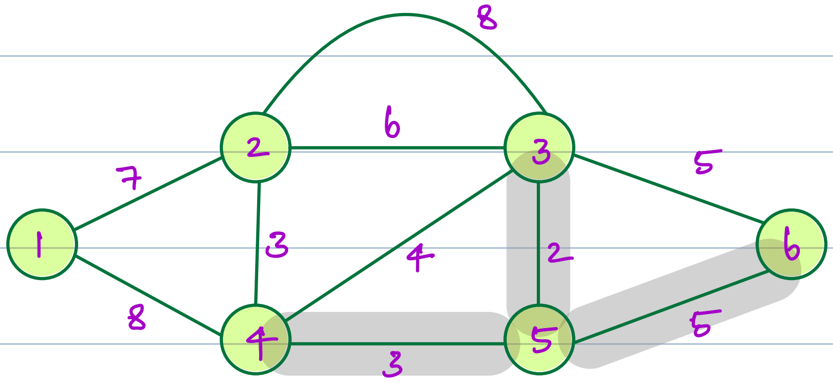
MST Algo

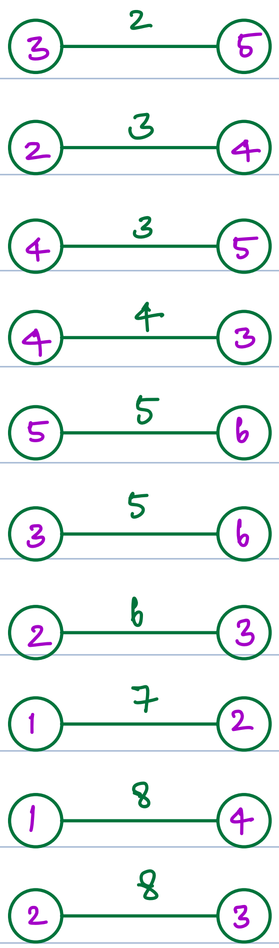
- Kruskal → DSA 4.2
- Prims

Prims Algorithm

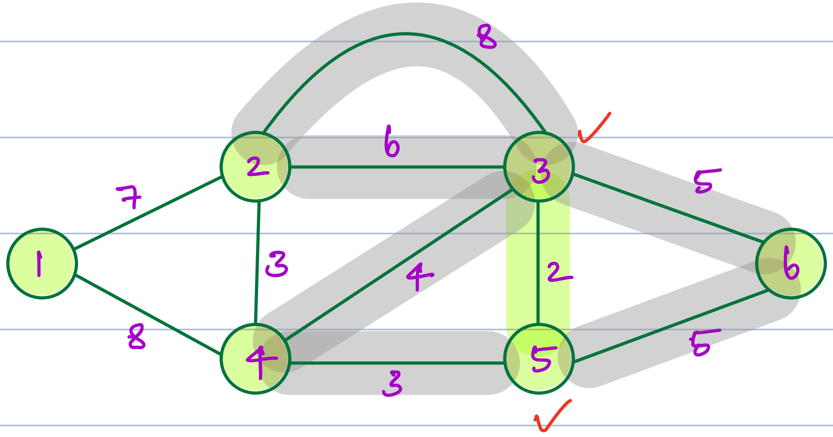


Starting at 5

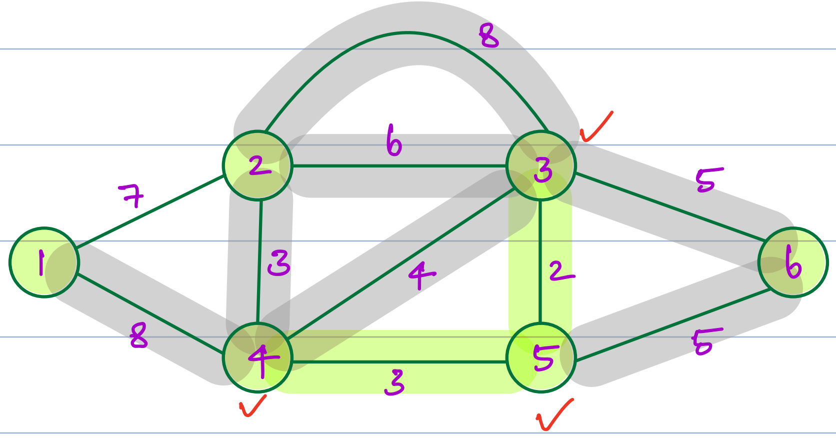




pick edge to 3 from 5

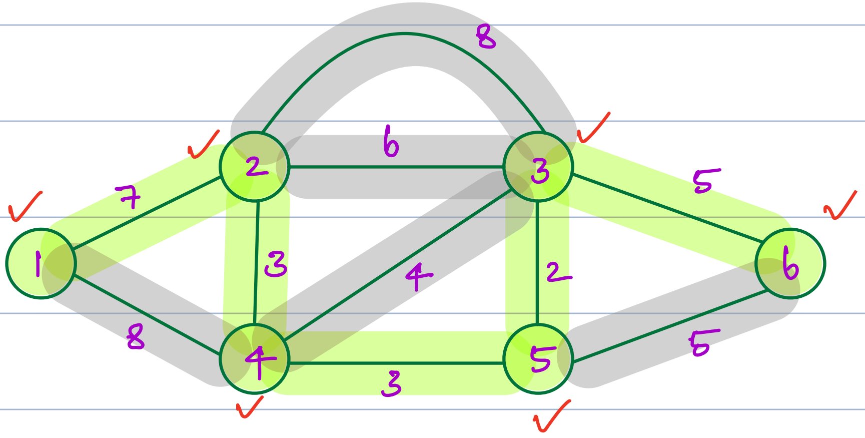


pick edge to 4 from 5



Quiz 1

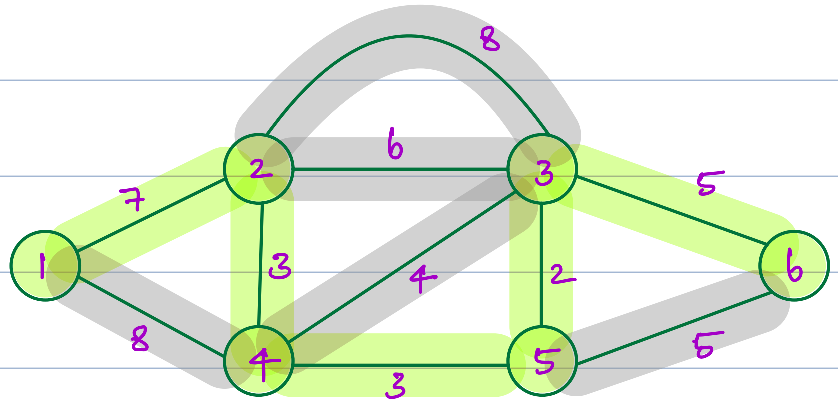
pick edge to 2 from 4



ans = 20

pick edge to 6 from 3

pick edge to 1 from 2



{ wei , nei }

Black box

~~(6km, 2)~~ ~~(8km, 2)~~ ~~(4km, 4)~~

~~(2km, 5)~~ ~~(5km, 6)~~ ~~(3km, 4)~~

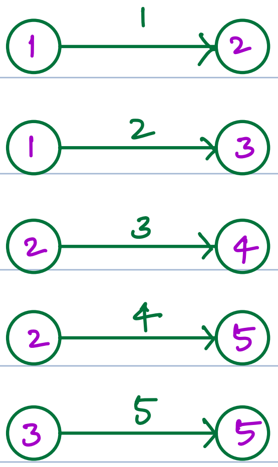
~~(5km, 6)~~ ~~(3km, 2)~~ ~~(8km, 1)~~

~~(7km, 1)~~

Visited array

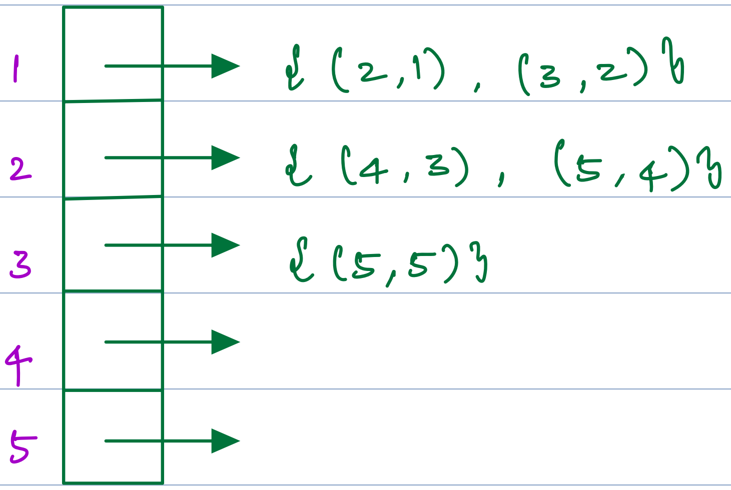
T	T	T	T	T	T
1	2	3	4	5	6

Quiz 2 : Min Heap for Black box



Adjacency list

{v, w}



1 → { (2,1) , (3,2) }

2 → { (4,3) , (5,4) }

3 → { (5,5) }

4 →

5 →

pseudocode

```
// v → nodes    e → edges
// graph → adj. list → T.C O(v+e)
// create a min heap
// visited [N] = {F, F, ...}
// given a start node
  for ({nei, w} in graph (start)) {
      heap.add ({w, nei});
  }

ans = 0;    visited [start] = True;
while (! heap.is Empty ()) {
      w, v = heap.get-min()
      if (visited [v]) continue;
      visited [v] = True;
      ans += W;
      for ({nei_n, nei_w} in graph [v]) {
            if (! visited [nei])
                heap.add ({nei_w, nei_n})
      }
}
```

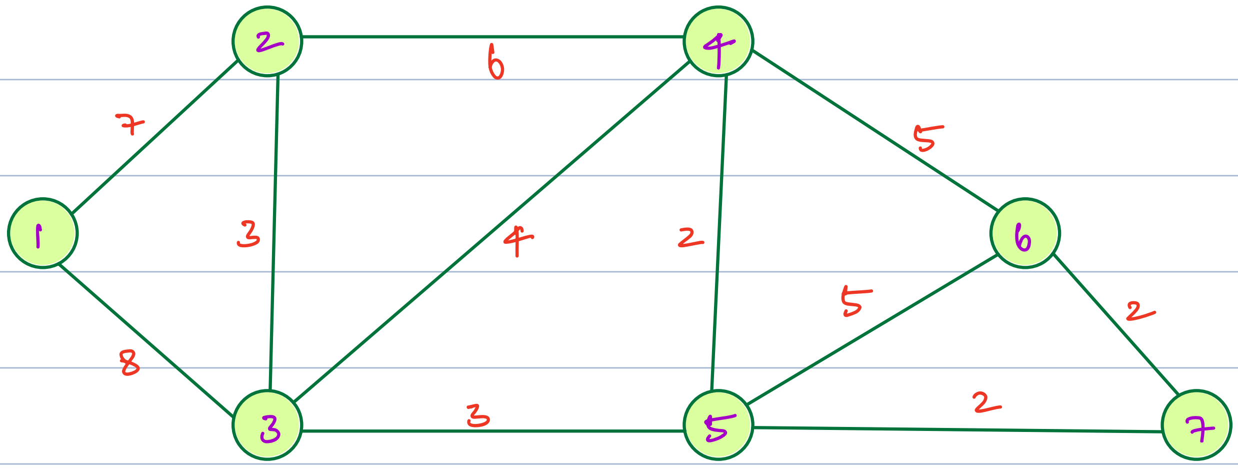
T.C = O(V+E + E log E).

10:14pm → 10:22pm

Dijkstra's Algorithm {single source shortest Path}

There are n cities in a country, you are living in city no. 1

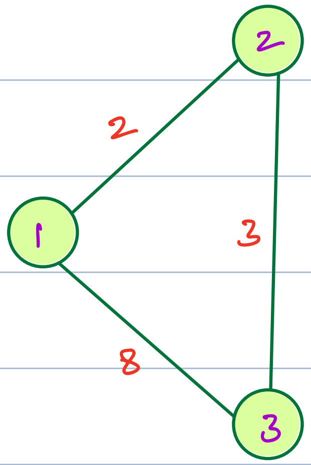
Find min distance to reach every other city from 1



Output →

0	7	.	.	.	.	.
1	2	3	4	5	6	7

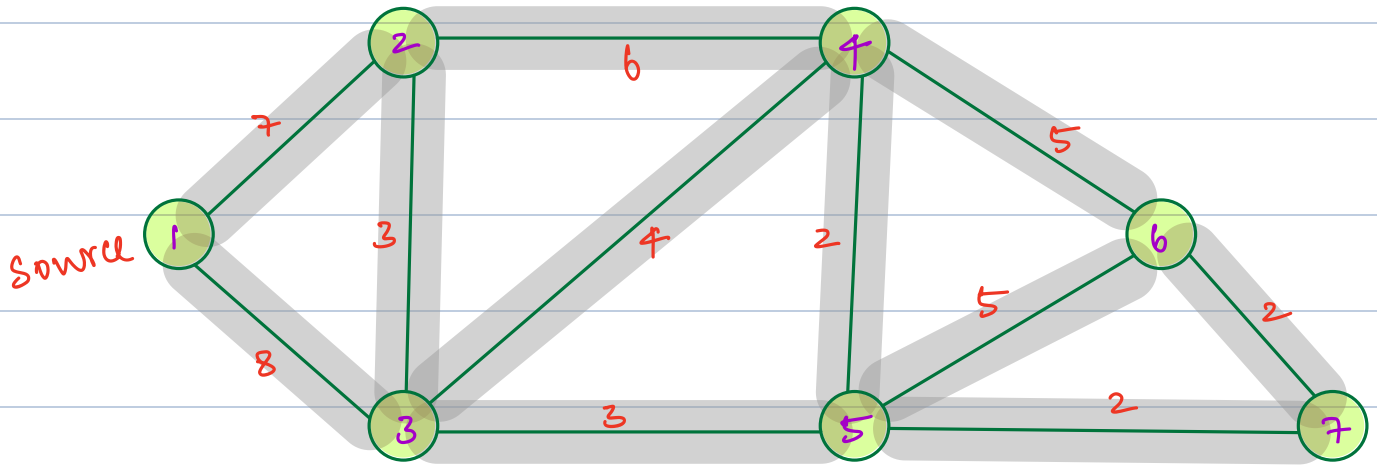
single source shortest path = Dijkstra



$\underline{D[1-3]} > \underline{D[1-2] + D[2-3]}$

Dijkstra will pick through 2

diferent from BST



distances =

0	7	8	12	11	15	13
~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
1	2	3	4	5	6	7

{ d, v }

~~(0,1)~~ ~~(7,2)~~ ~~(8,3)~~ ~~(10,3)~~ ~~(13,4)~~

~~(12,4)~~ ~~(11,5)~~ ~~(13,4)~~ ~~(13,7)~~

~~(16,6)~~ ~~(17,6)~~ ~~(15,6)~~

Min Heap

Black box

Quiz 3

pseudo code

```
// graph ——→ Adjacency list
// create min heap
// distance [n] = { INT_MAX , INT_MAX ... }
heap . add (0, start) ;
distance [start] = 0;
```

```
while (! heap.isEmpty()) {
        d, n = heap.getmin();
        if (d > distance[n])   continue;
         for ({nei, w} in graph[n]) {
              int n_dist = distance[n] + w;
              if (n_dist < distance[nei]) {
                    distance[nei] = n_dist;
                    heap.add({n_dist, nei});
              }
         }
}
```

T.C = O(E log E

Quiz 4:

Announcements:

Revision Material

Brute Force / partial

Assignments (Topics you are Doubtful)