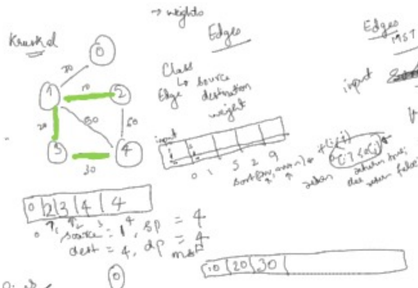
Order:
→ ~~Kruskal Imp~~
→ ~~Prim's Logic~~
→ ~~Prim (MST) → Dij (K,D)~~
→ Prim's Algo
→ Dijkstra
→ Imp.

Kruskal



→ weight

Edges

Check Edge source destination weight

input

1	2	5	2	9
0	1	2	3	4

Edges MST

input

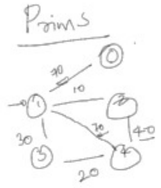
→ 1 2 10
→ 1 3 20
→ 3 4 30
→ 1 4 50
→ 2 4 60
→ 0 1 70

0	1	2	3	4	4

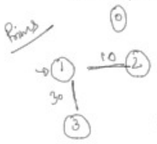
source: 1, sp = 4
dest = 4, dp = 4
mst

10	20	30

Prims

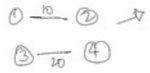


Prims



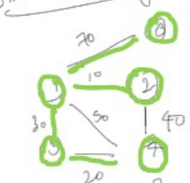
→ If more than one MST possible, some edge weights must be repeating.

Kruskal



Con.

Prims Algo



priority queue
min-heap

	0	1	2	3	4
key →	∞	10	0	∞	40
parent	-1	2	0	-1	2

visited: 2, 1, 3, 4

2, 0, 0 — $\boxed{\log V}$

	0	1	3	4
	∞	10	∞	40
	-1	2	-1	2

1, 10, 2 — $\boxed{\log V}$

	0	3	4
	70	30	40
	1	1	2

3, 30, 1 — $\boxed{\log V}$

	0	4
	70	20
	1	3

4, 20, 3 — $\boxed{\log V}$

2, $\log V$
2, $\log V$
3, $3 \log V$

n → O(n)

find min V times — setting up — visiting edges — updating others on behalf of the min (E)

$$O(V\log V + V + 2E + E\log V) \leftarrow \text{Min Heap \& Adj List}$$

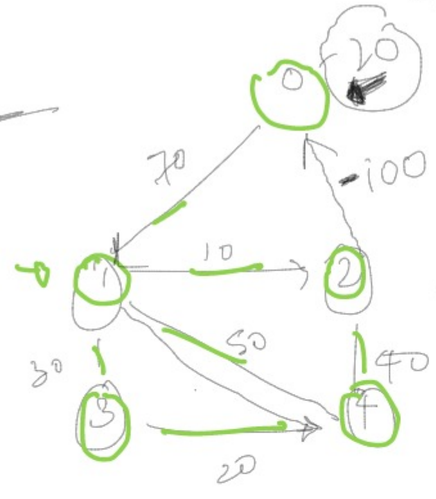
$$\rightarrow O((V+E)\log V) \rightarrow O(E\log V)$$

Min Heap & Adj Matrix

$$O(V\log V + V + V^2 + E\log V)$$

$$\rightarrow O(V^2 + E\log V)$$

Dijkstra



	0	1	2	3	4
dist →	0	∞	∞	∞	∞
parent →	0	-1	-1	-1	-1
		70	80	100	120
		0	1	1	1

0, 0, 0

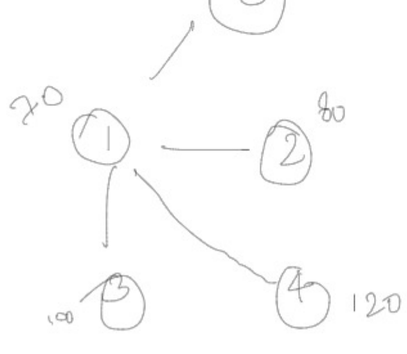
1, 70, 0

2, 80, 1

3, 100, 1

4, 120, 1

relaxing vertices only once, -ve edge weight cycles



Dijkstra