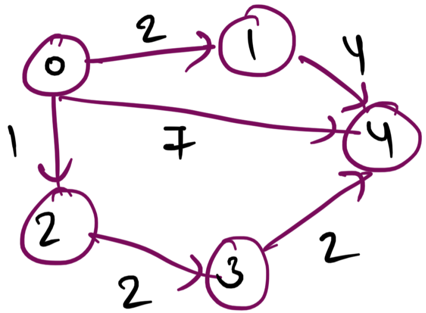
Graphs - IV

Shortest Path in a Directed Acyclic Graph



Topological sorting

0	1	2	3	4

$u \to v$

src = 0

0	2	1	3	5
0	~~∞~~ 2	~~∞~~ 1	~~∞~~ 3	~~∞~~ ~~7~~ ~~6~~ 5
0	1	2	3	4

dis[] =

$\infty > 0 + 2$

$\infty > 0 + 1$

$\infty > 0 + 7$

$7 > 2 + 4$

$\infty > 1 + 2$

$6 > 3 + 2$

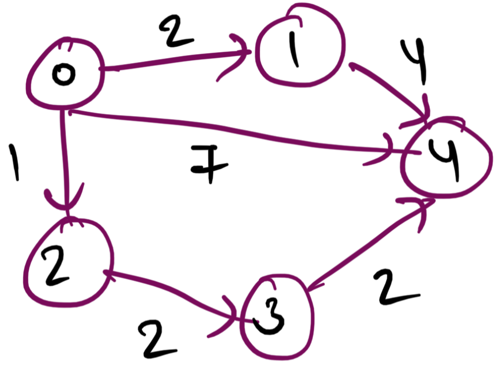
$\boxed{O(V+E)}$

```
for every neighbor (v) of u: {
    if ( dis[v] > dist[u] + wt (u,v) ) {
        dist[v] = dist[u] + wt(u,v);
    }
}.
```

(v, wt)

- 0 → (1,2) | (2,1) | (4,7)
- 1 → (4,4)
- 2 → (3,2)
- 3 → (4,2)
- 4

	0	1	2	3	4
0		2	1		7
1					4
2				2	
3					2
4					



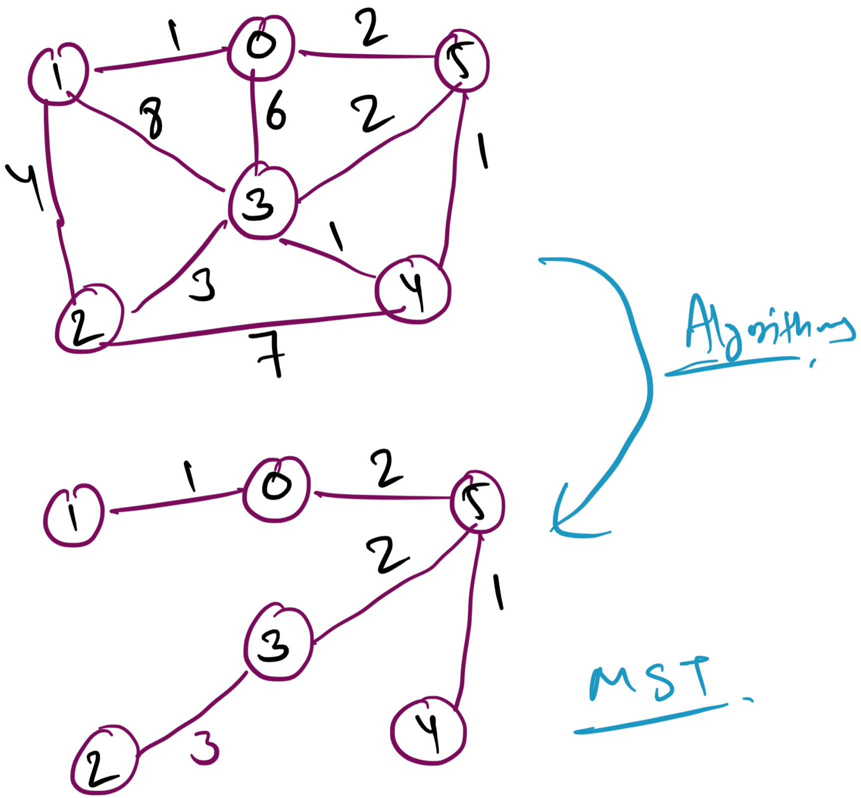

```
class Pair {
    int v
    int wt
}
```

Minimum Spanning Tree

\# Minimum edge sum graph such that each vertex can be visited by another vertex.

V vertex

MST $\longrightarrow$ (V-1)

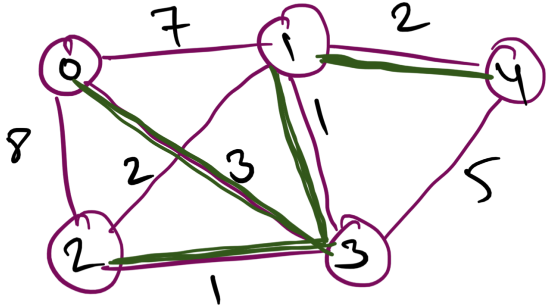


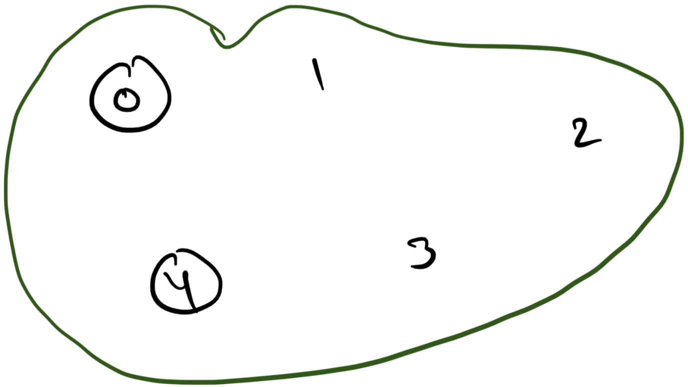
Kruskal

[redacted]al Algorithm for MST

Greedy Algorithm

1. Sort the edges by their weights.





V = 5
(V−1)

✓ (1, 3) → 1
✓ (2, 3) → 1
× (1, 2) → 2
✓ (1, 4) → 2
✓ (0, 3) → 3
(3, 4) → 5
(0, 1) → 7
(0, 2) → 8

$E \log E + E$

Prim's Algorithm for MST

MinHeap → Pair (wt) .

Pq. →

(1, 7) X(2, 2)

X(3, 3)

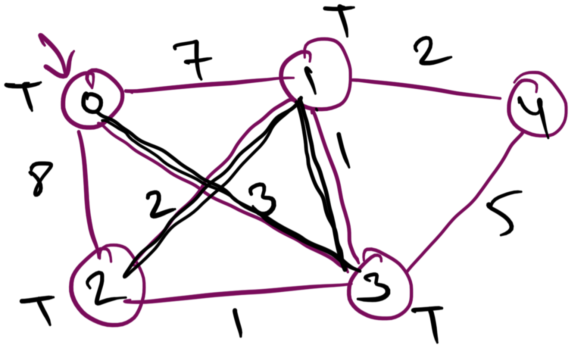
(2, 8)

X(1, 1)

X(2, 1)

(4, 5)

(4, 2)



class Pair {
int v
int wt;
}.

ans → (0, 3), (1, 1), (2, 1)

0

pq

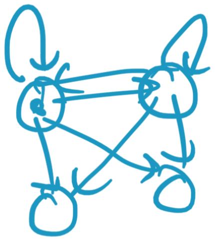
- (3, 3)
- (1, 7)
- (2, 8)
- ~~(1, 1)~~
- (2, 1)
- (4, 5)
- ~~(4, 2)~~

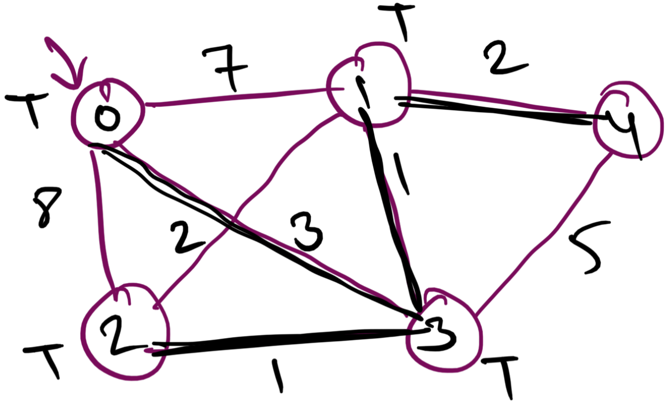
pq + hashmap.

$\boxed{O(E)}$

$\boxed{O(V)}$

code.





dis

0	1	2	3	4
0	~~7~~ 1	~~8~~ 1	3	~~8~~ 2

vis

T	T	T	T	–

curr = (4, 2)

$\boxed{V \log V}$

V^2

$\log V^2 \rightarrow \log V$

~~(1, 7)~~

~~(2, 8)~~

~~(3, 3)~~

~~(1, 1)~~

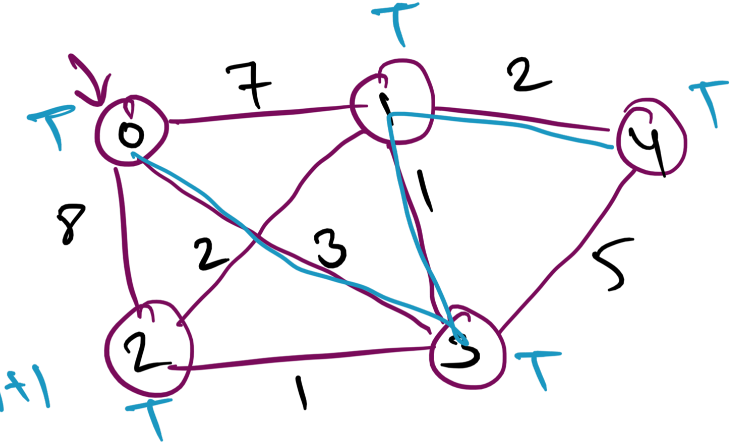
~~(2, 1)~~

~~(4, 5)~~

~~(4, 2)~~

$E \uparrow$

$O(\boxed{E \log E}) = \boxed{E \log V}$



ans = 0+3+1+1 +2

$\underline{7}$.

0	1	2	3	4
0	~~7~~ 1	~~8~~ 1	3	~~8~~ 2

$E << V$

$\log E \longrightarrow \log V$

Dijkstra's Algorithm for Shortest Distance in a Weighted Graph

pq

~~(1, 1)~~

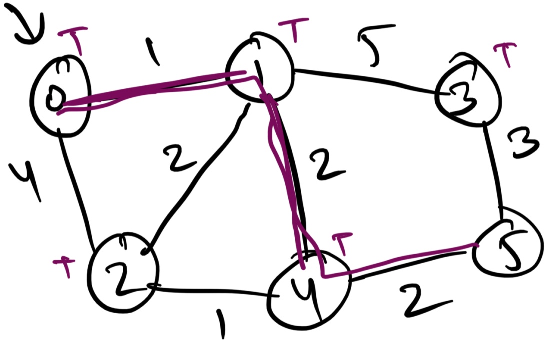
~~(2, 4)~~

~~(2, 3)~~

~~(4, 3)~~

~~(3, 6)~~

~~(5, 5)~~



dis

T	T	T	T	T	T
0	1	~~4~~3	6	3	5
0	1	2	3	4	5

0 1 3 6 3 5

par	-1	0	1	1	1	4
	0	1	2	3	4	5

```
int cur = 5
while (par[cur] != -1){
        print (cur)
        cur = par[cur]
}
```

Bellman Ford Algorithm for Shortest Distance in a Weighted Graph