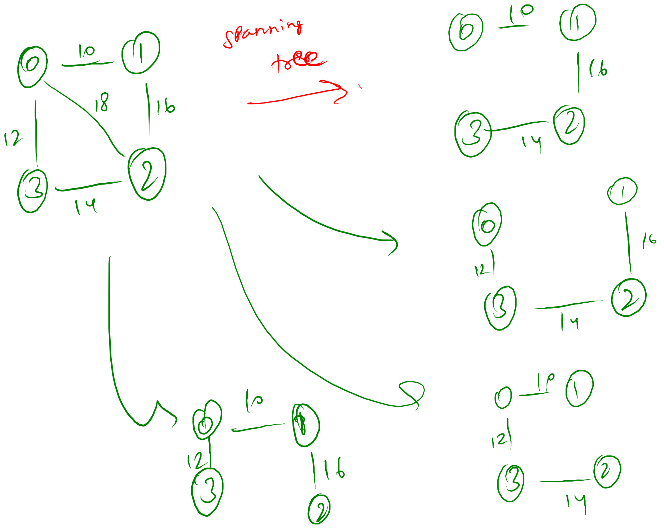
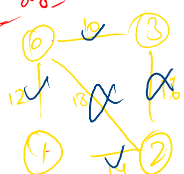
MST



Prim's algo



V vtx → (V-1) edges

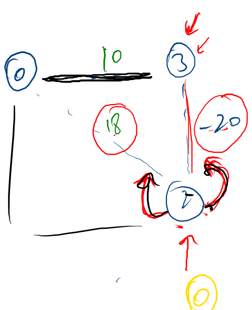
root vtx = 0

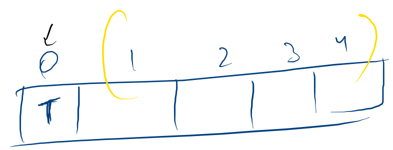
minimum

vtx1 → edge 1

vtx2 → edge 2

vtx3 → edge 3





Shortest Path algo → directed, undirected → +ve, -ve, 0

edge count ← bfs → $O(V+E)$

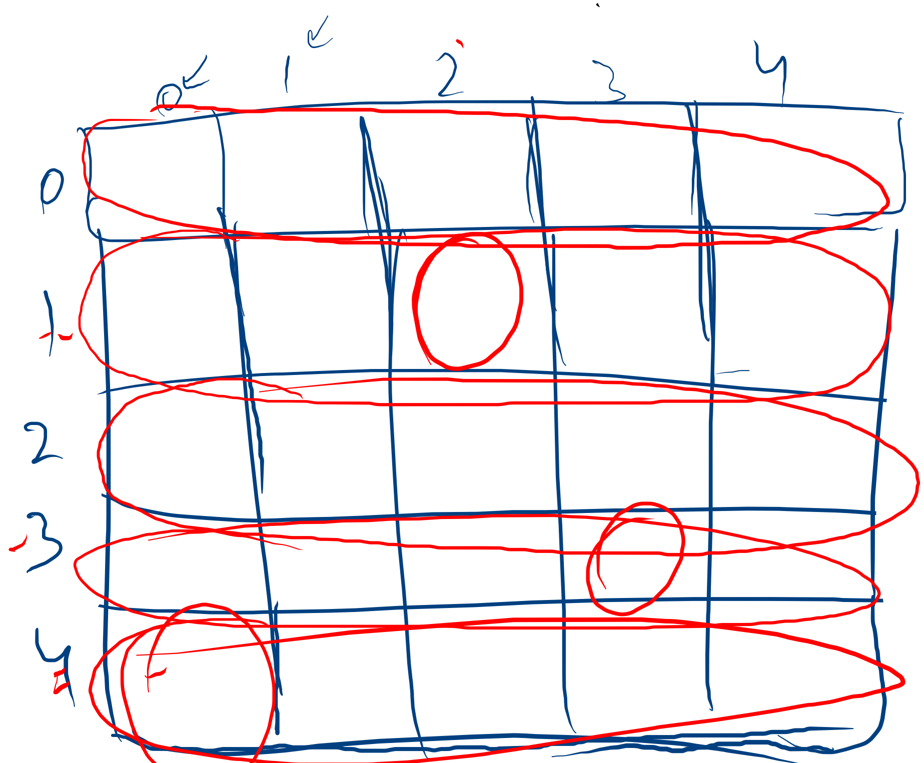
edge wt

edge wt

single src shortest path
- dijkstra → $O(E \log V)$ → allowed → (0, +ve)
- bellman ford → $O(E+V)$ → allowed → (0, +ve, -ve)

All pair shortest path
- floyd warshall → $O(V^3)$ → allowed → (0, +ve, -ve)

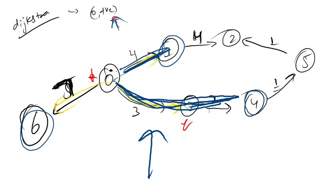
src → fin

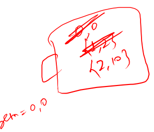


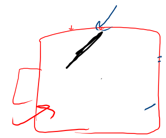
-ve wt cycle

↓

shortest αα



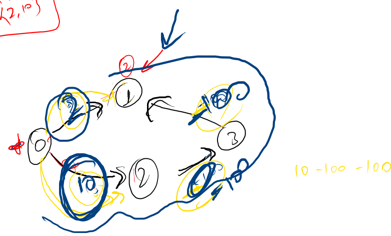


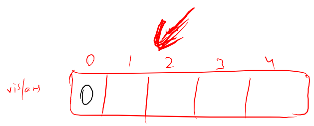


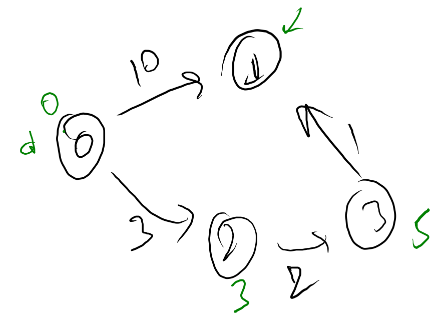
E log E

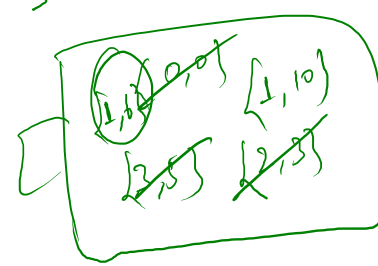
⇓

E log V







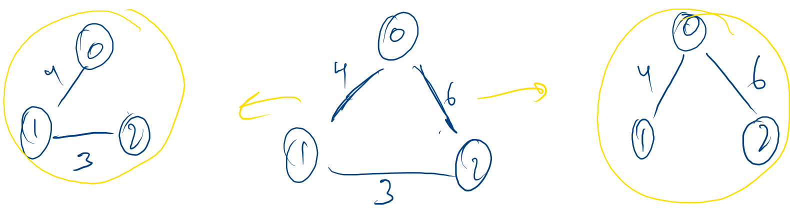


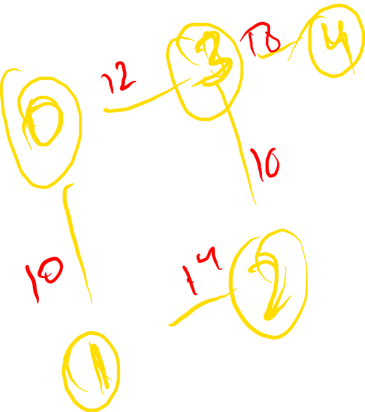
Prim's | dijkstra

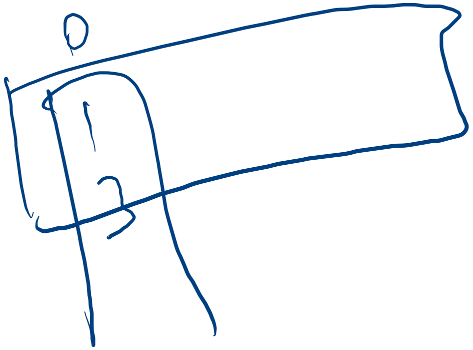
→ overall graph minimize | Path minimize

⇓ | ↓

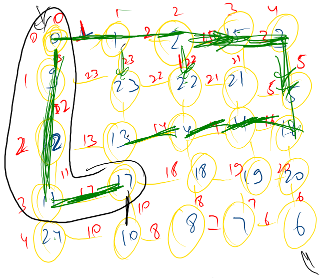
+ve, -ve, 0 | +ve, 0







f \ t	0	1	2	3	4
0	0	10	0	12	0
1	10	0	14	0	0
2	0	14	0	16	0
3	12	0	10	0	18
4	0	0	0	18	0



```java
    public int swimInWater(int[][] grid) {
        PriorityQueue<Pair> pq = new PriorityQueue<>();
        pq.add(new Pair(0, 0, grid[0][0]));
        boolean[][] vis = new boolean[grid.length][grid.length];
        int[][] dirs = { { -1, 0 }, { 0, 1 }, { 1, 0 }, { 0, -1 } };

        while (pq.size() > 0) {
            Pair rem = pq.remove();
            if (rem.row == grid.length - 1 && rem.col == grid[0].length - 1) {
                return rem.msf;
            }
            if (vis[rem.row][rem.col] == true) {
                continue;
            }

            vis[rem.row][rem.col] = true;

            for (int i = 0; i < dirs.length; i++) {
                int rowdash = rem.row + dirs[i][0];
                int coldash = rem.col + dirs[i][1];

                if (rowdash < 0 || coldash < 0 || rowdash >= grid.length || coldash >=
grid[0].length
                        || vis[rowdash][coldash] == true) {
                    continue;
                }

                pq.add(new Pair(rowdash, coldash, Math.max(rem.msf, grid[rowdash]
[coldash])));
            }
        }

        return 0;
```

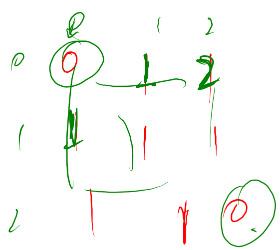
4^n

$\left[0 \quad \overset{mid}{\frac{n^2-1}{2}} \quad \right] \quad n^2-1$

7

{1, 2, 3}

{0, 0, 0} ~~{0, 1, 1}~~ {1, 1, 2} {1, 1, 2} ~~{1, 0, 1}~~ {2, 0, 2}



0-1 matrix

↳ add

bfs

add

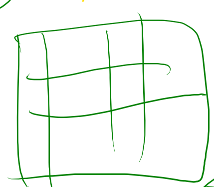
remove

Prim's

dijkstra

remove

n^2



$(0$

mid

$\frac{n^2-1}{2}$

mid

$0 \quad n^2-1$

$(\quad)$

$n^2 \log n^2$

N=4 C=28 +10 +10 +10

1 2 3 4 5 6 7

{15, 25, 3, 14, 10, 16, 9}

40 33 10 9

2

$\frac{40}{2}$ = 20

4 6

$\frac{33}{3}$ = 11

$\frac{10}{1}$

10

$\frac{9}{1}$

9

{3, 4}

{4, 6}

{1, 2}

36 − 33 = 3

1 + 1 + 3