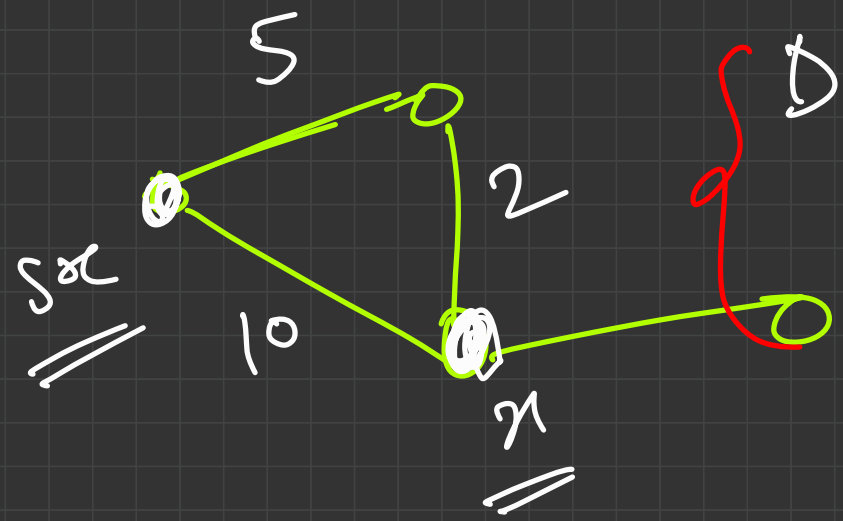
Shortest path algorithms

Graphs 2

Bipartite Graphs, Dijkstra, Bellman Ford

≥ 0

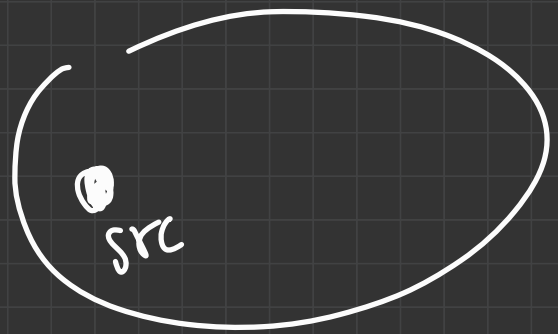
-Priyansh Agarwal



Dijkstra

Bellman ford

≥ 0



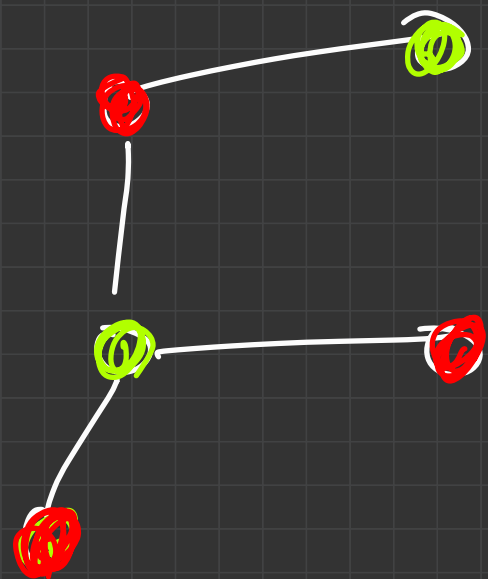
Bi-partite Graphs

- Definition
- Algorithm
- Odd Length Cycle → not bi-partitely colourable
- Tree Property
- Problem: Link

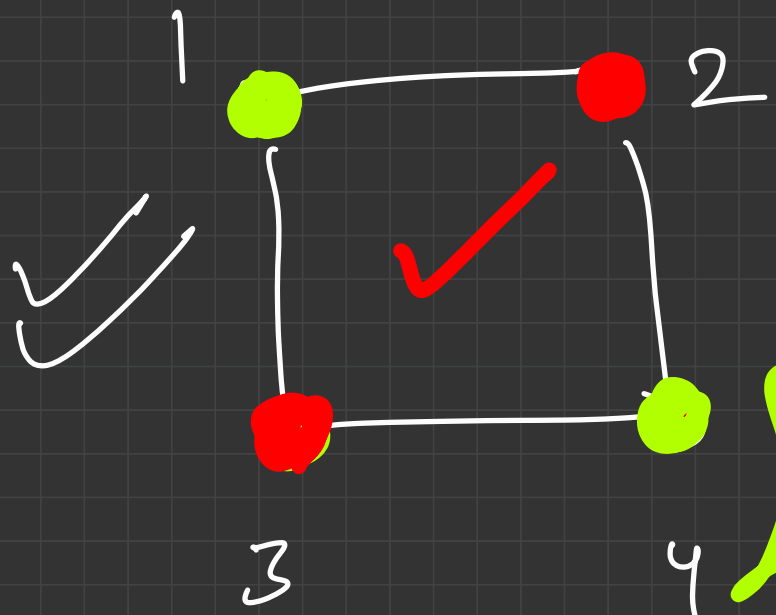
H.W

Bipartite Graphs :

Every 2 adj. nodes have different colors

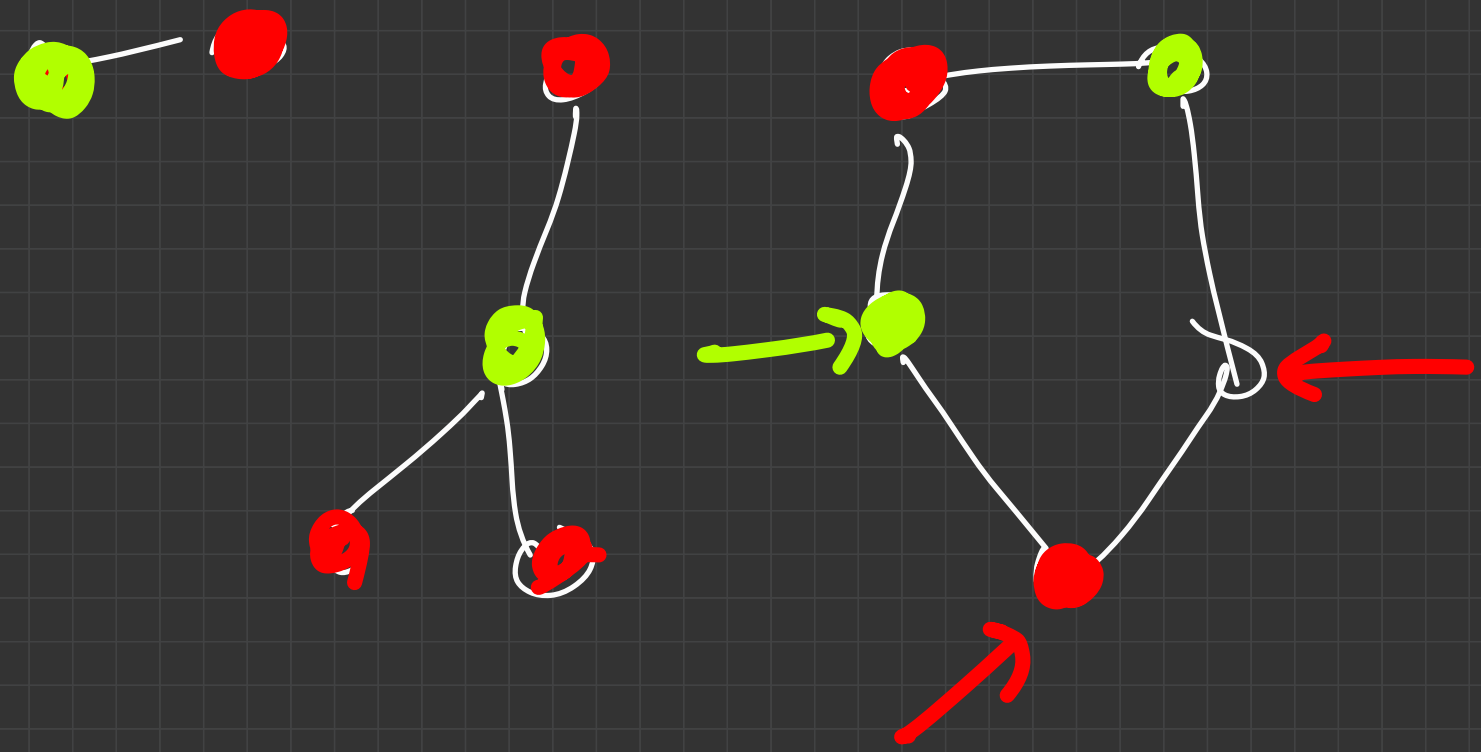


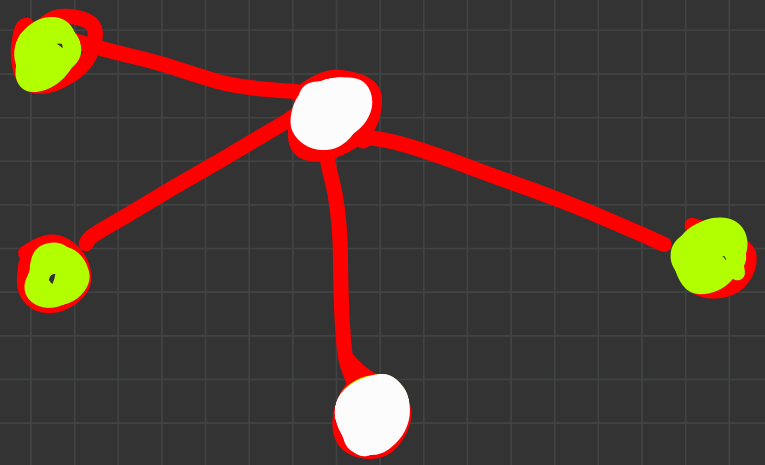
2 colors



R G

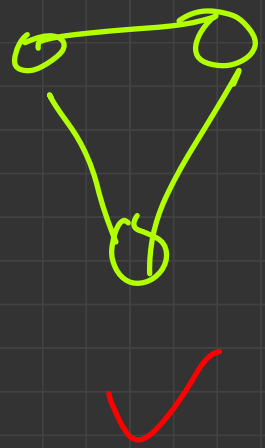
Given a connected graph it can only have 2 possible colorings (bi-partite)

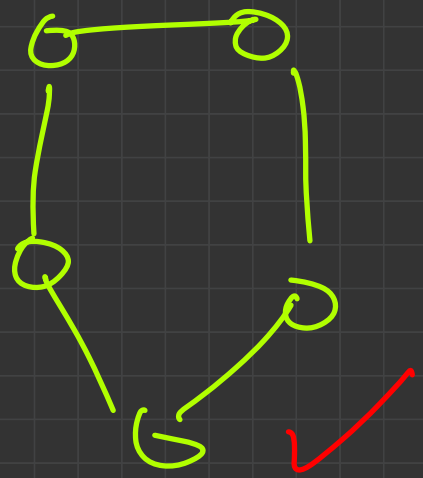




1

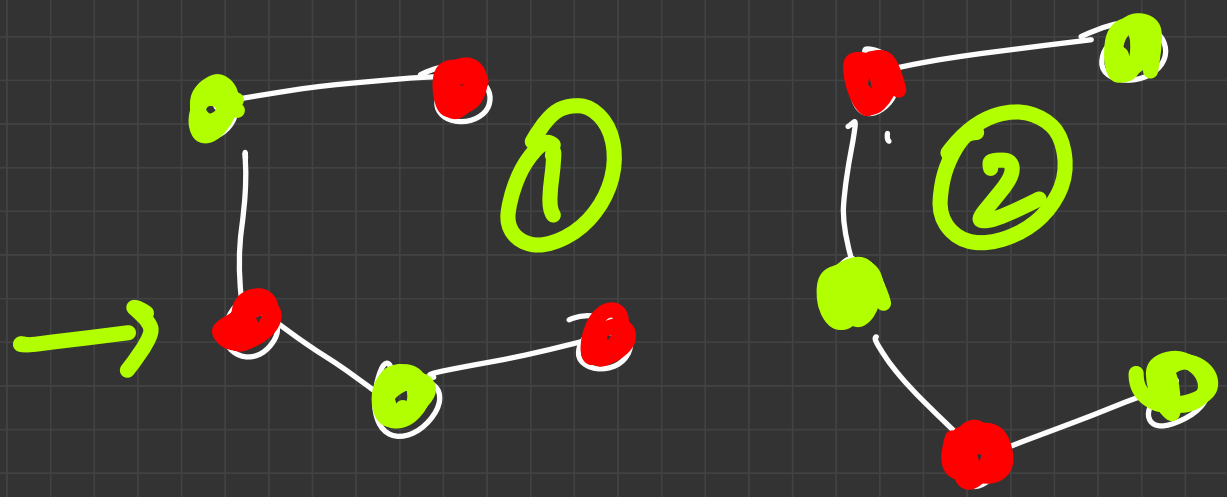
A graph can be colored in a bi-partite manner if and only if it does not contain an odd length cycle



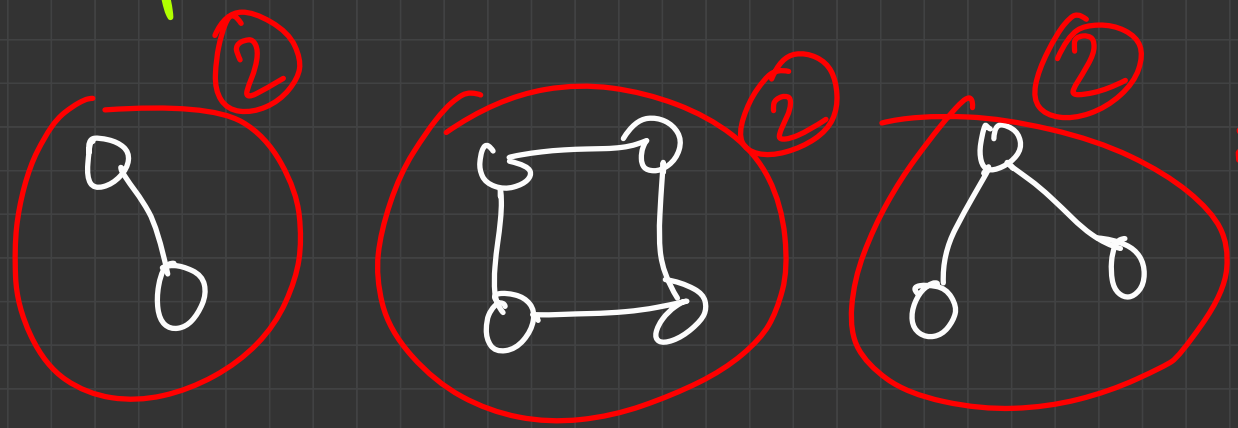


A connected graph can have either

0 bipartite colorings or 2

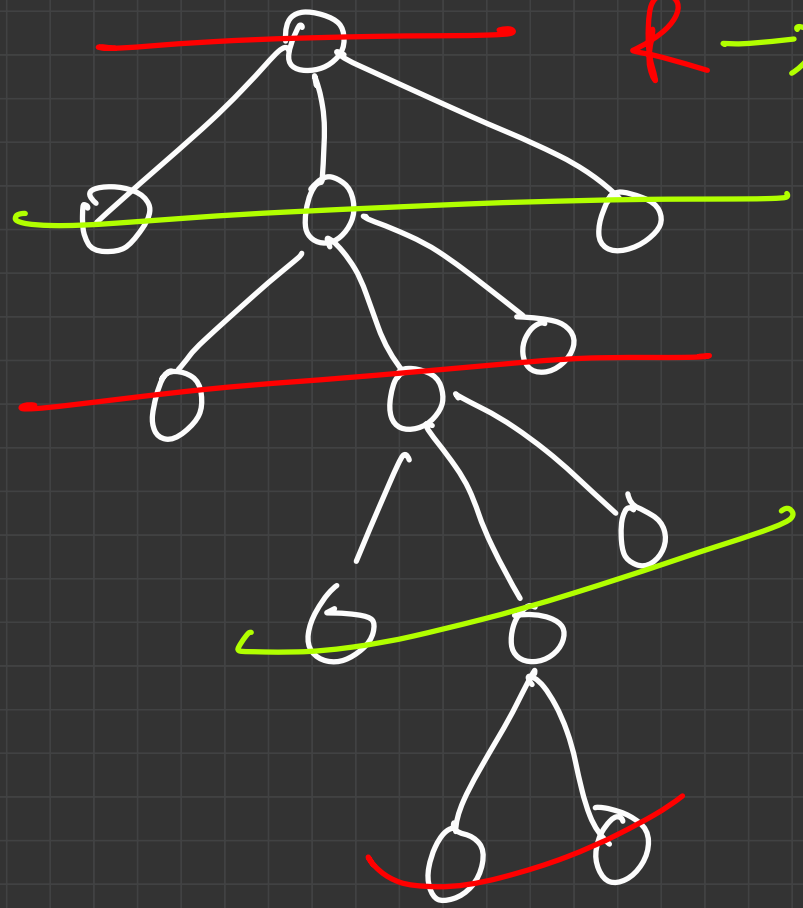


If a graph contains x connected components how many bi-partite colorings can you have



$= \underline{\underline{2^x}}$

2



R → G

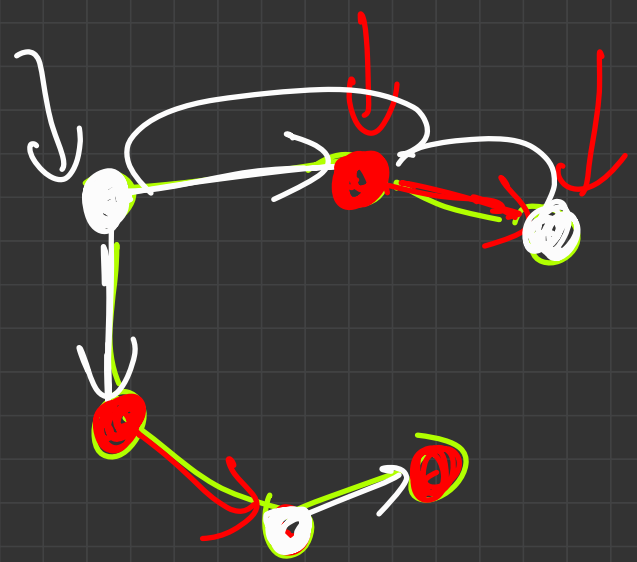
G → R

R → G

G → R

R → G

Choose a random node and color it with first color



```
vector <int> colors (n,0);

bool dfs (int curr, edges, colors)
{ bool ans = true
  for (int neighbour : edges[curr])
      if ( colors(neighbour) ==0)
          colors (neighbour) = 1 if curr color = 2
                               2 if curr color = 1
          ans = ans && dfs( neighbour, edges, colors)
      else if ((colors(neighbour) == colors(curr))
```

W = 1
R = 2

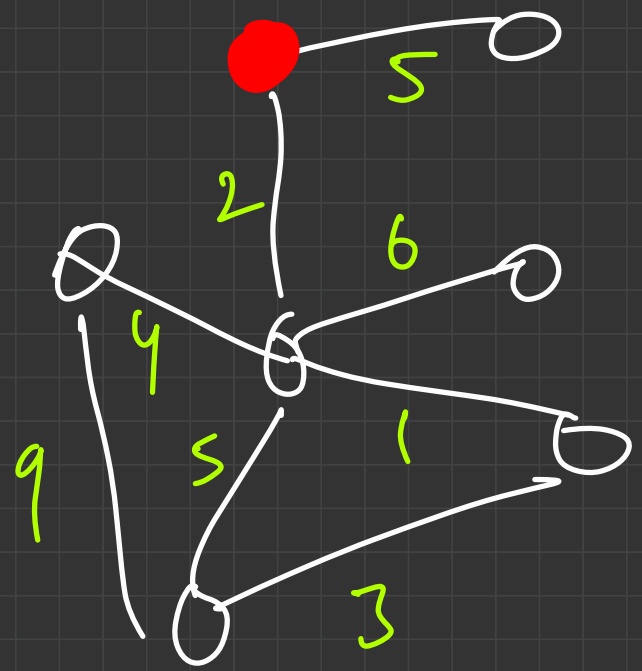
```
        return false;
    return ans;
}
```

$O(n)$

```
bool ans = true
for( int i=0 ; i<n ; i++) {
    if (colors[i] ==0)
        colors[i] = 1
    ans = ans && dfs(i, edges, colors) }
```

Dijkstra (most important)

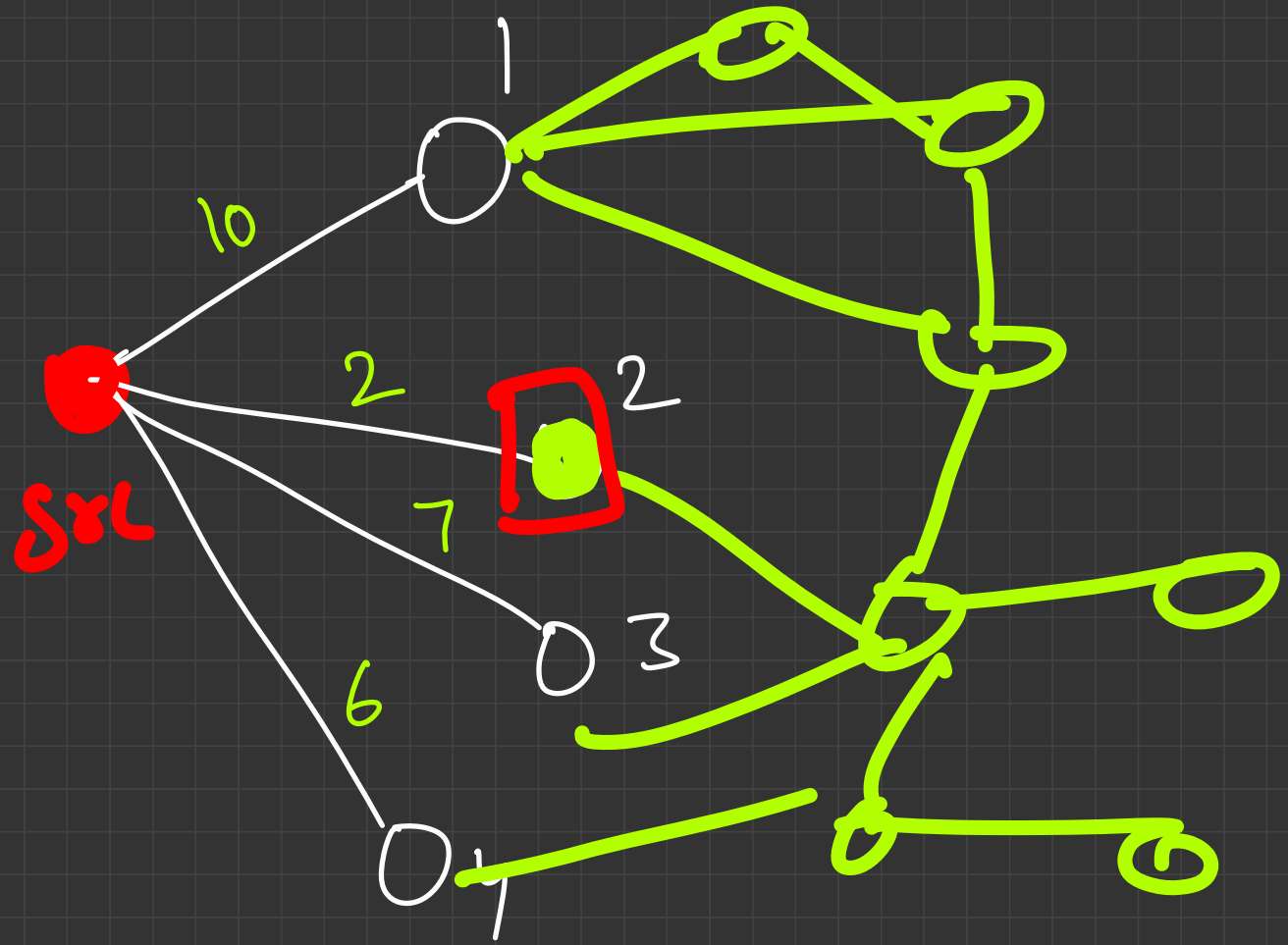
- Single Source Shortest Path Algorithm - Idea + Visualization
- Non-negative edge weights
- Proof/Intuition:
    - On every iteration the marked vertex is the one that can never have a better distance later on.
- Code
- Retrieving the shortest path?
- Problems
    - Google Interview Problem

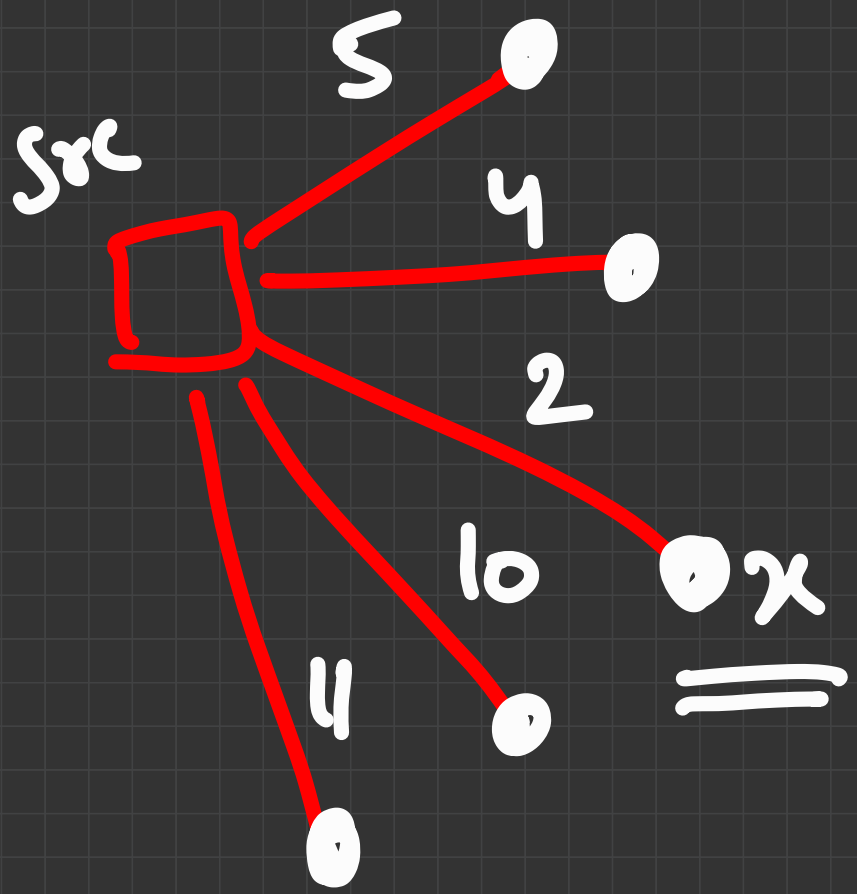


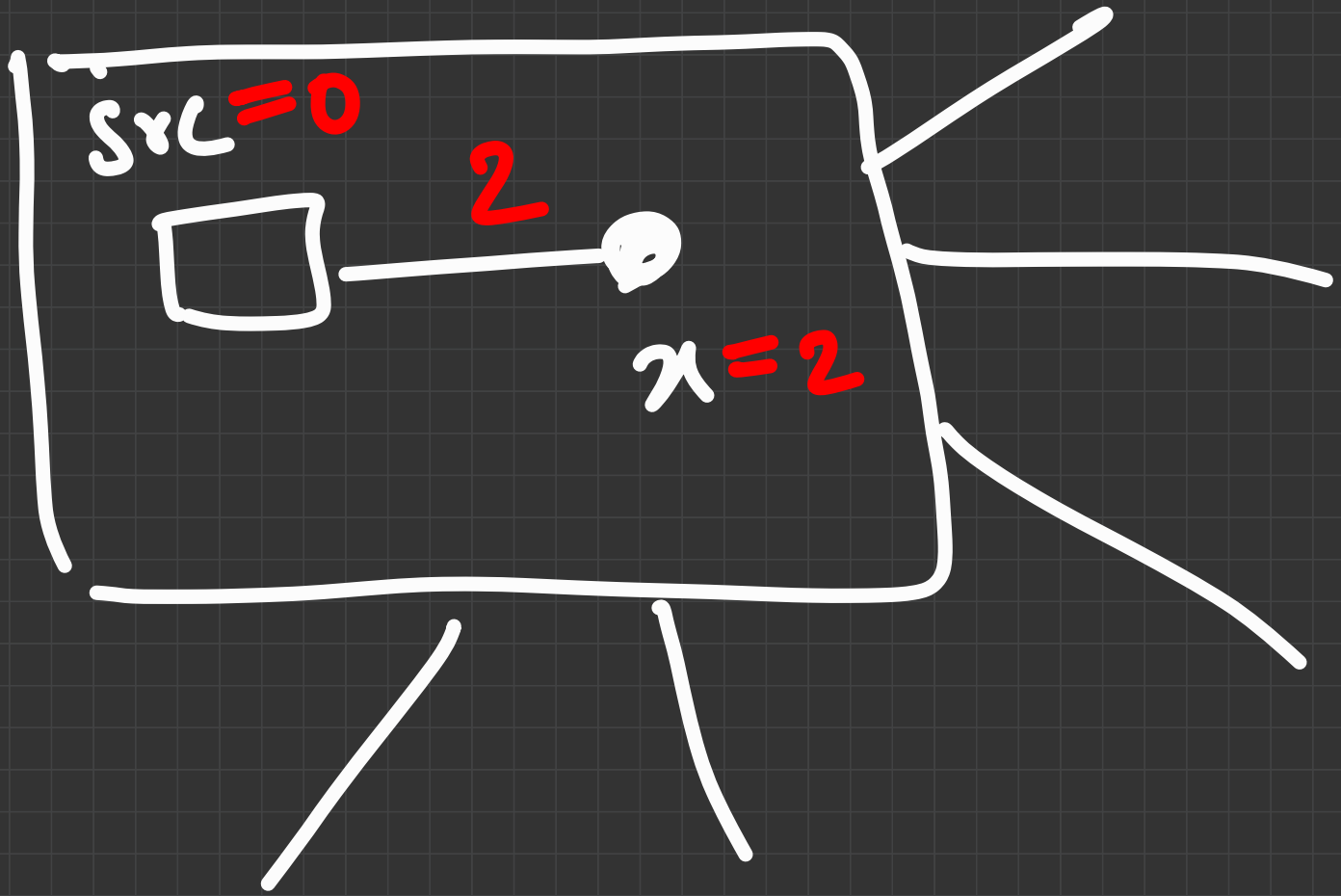
all edge weights ≥ 0

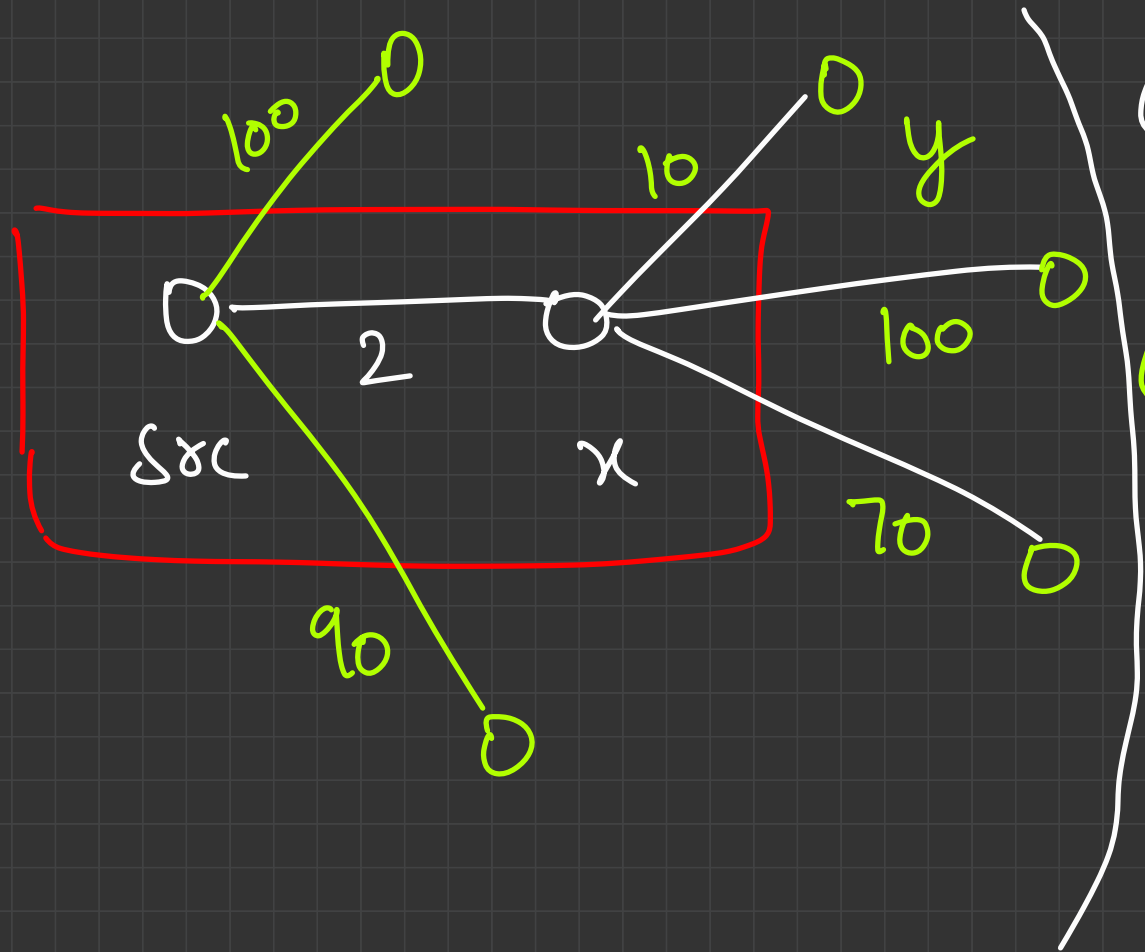
non-negative

Src


1
10
2
2
S&L
7
3
6
4


Src
5
4
2
10
x
11

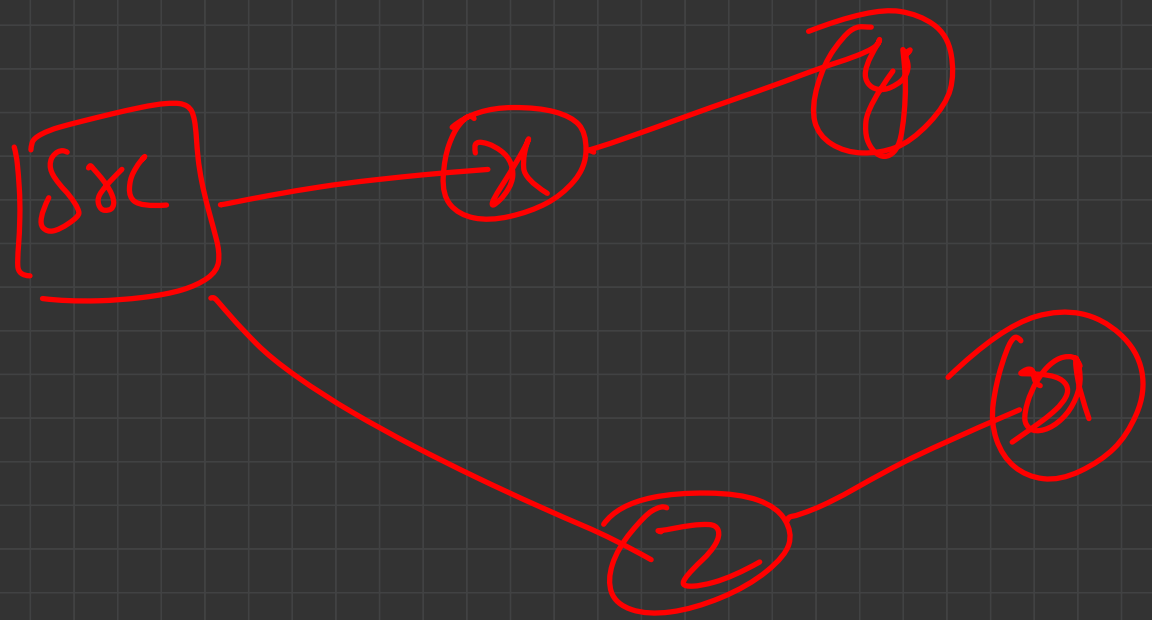

SxC=0
2
x=2

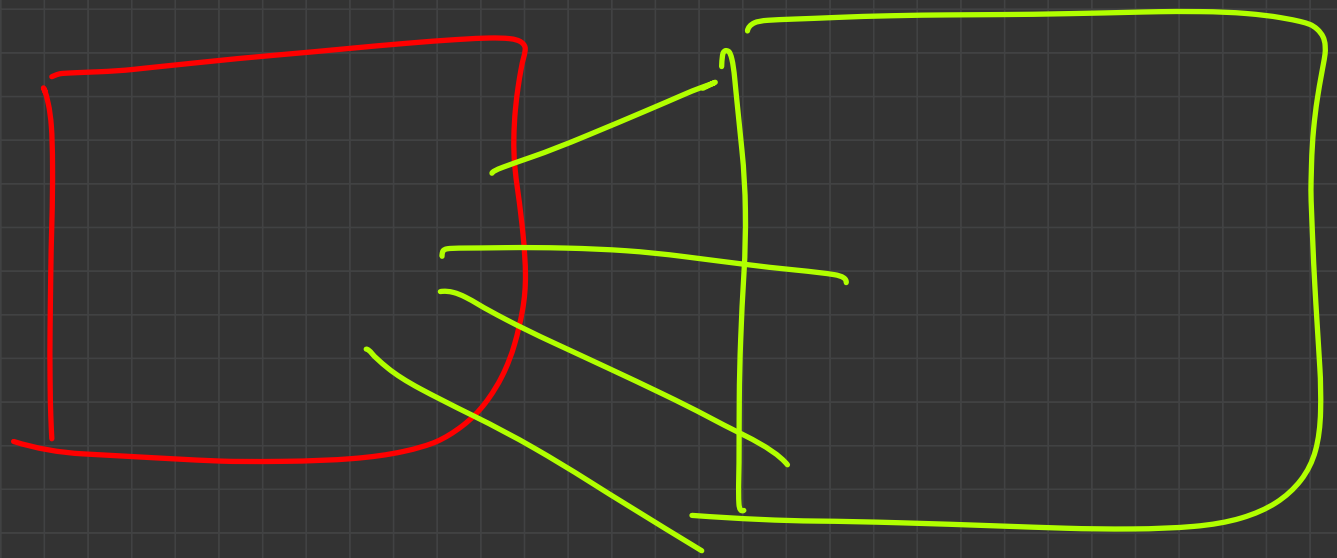


① Coming out of x

② Coming out of src

12

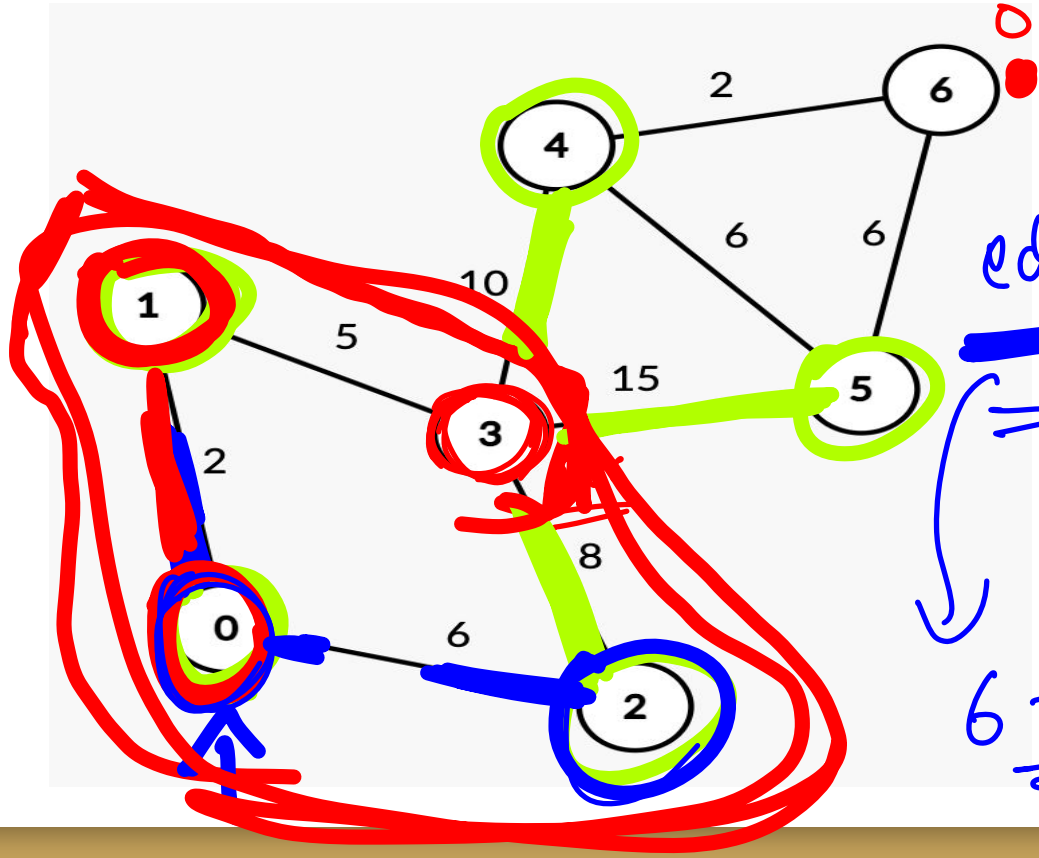

Src
x
y
z
w



Dijkstra Visualization

dist

0	1	2	3	4	5	6
7	5	8	0	10	15	inf



edge weight + dist[u]

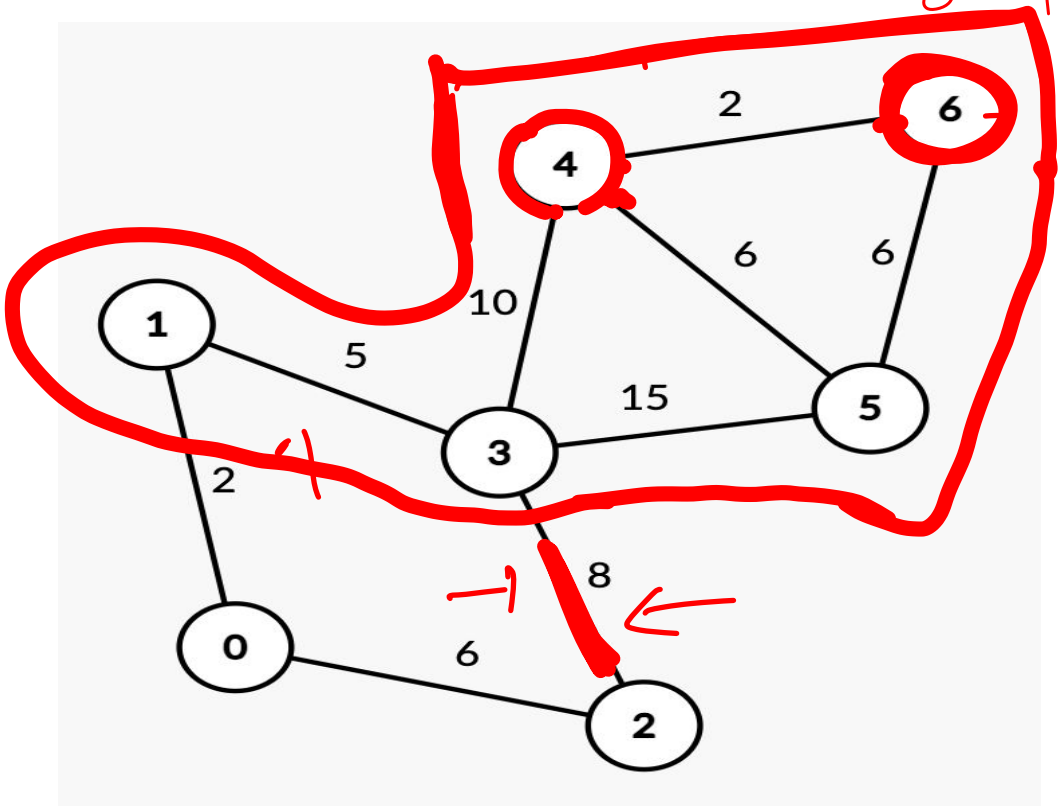
= 2 + 5 = 7

6 + 7 = 13

Dijkstra Visualization

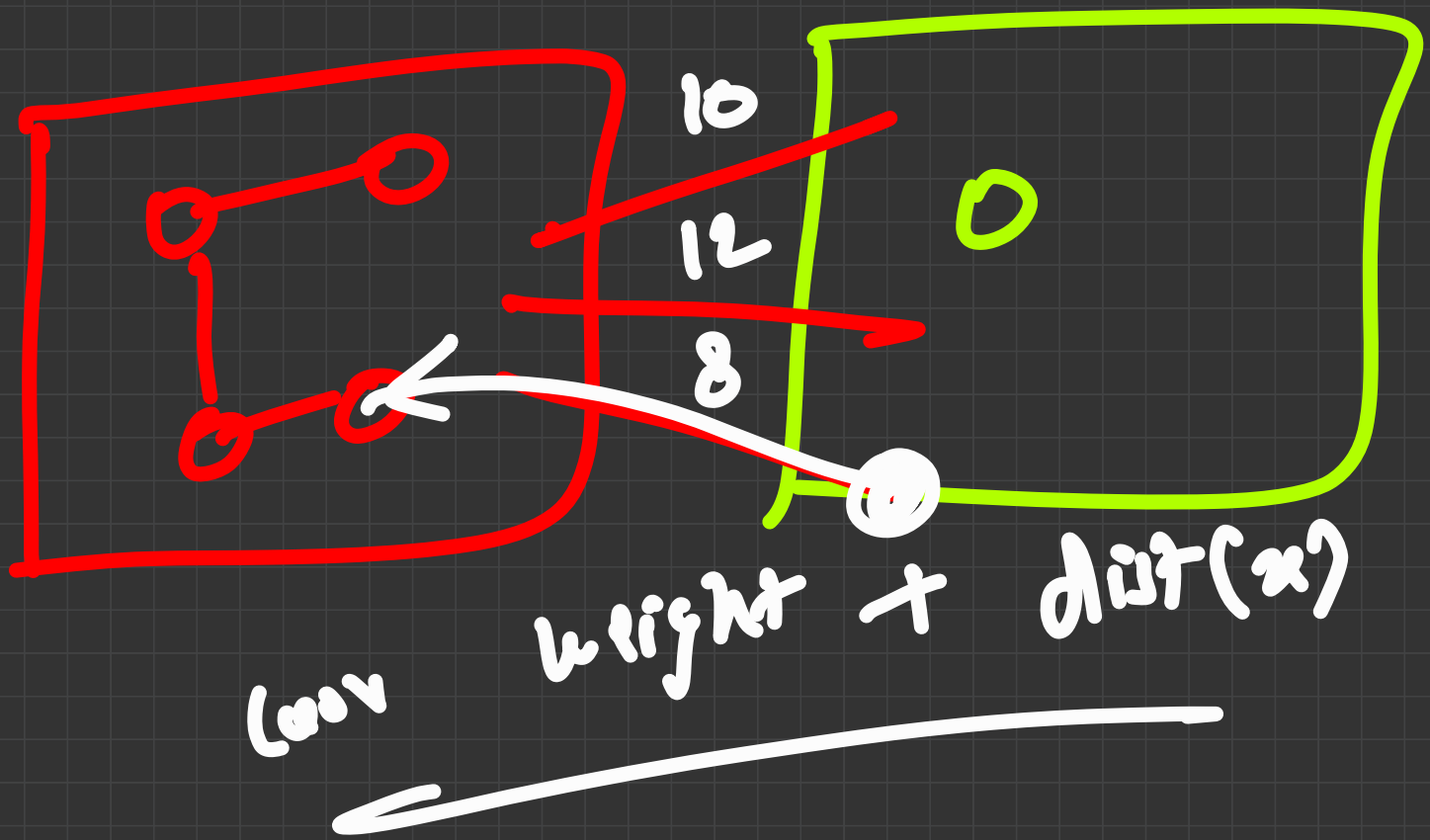
dist

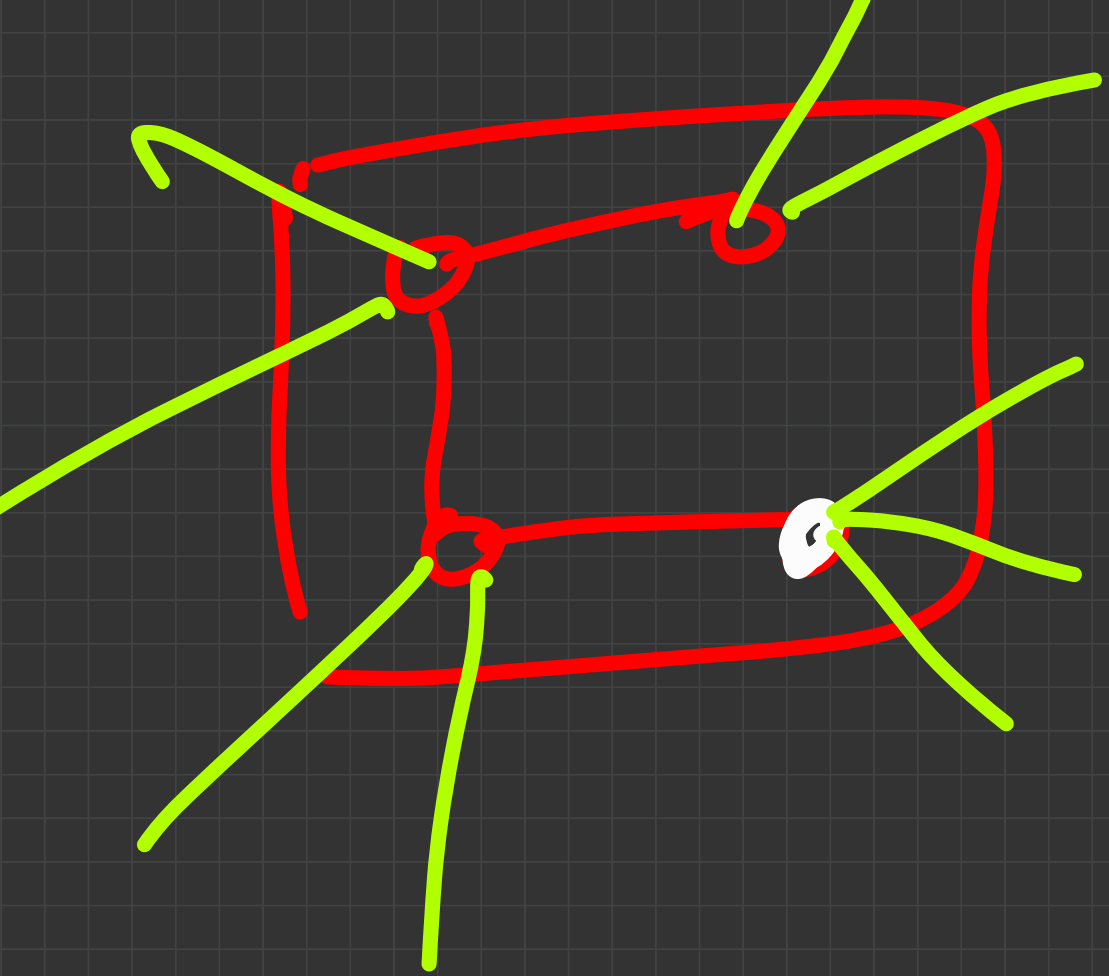
10^9	15	18	10	0	6	2
0	1	2	3	4	5	6



5 + 10

8 + 10


10
12
8
0
low weight + dist(x)

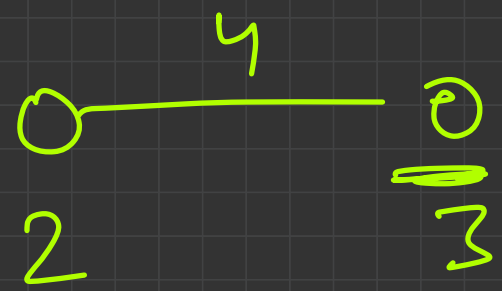


$$\boxed{(14,3),\ (15,3),\ (20,4)}$$

Set

$(20,2)$

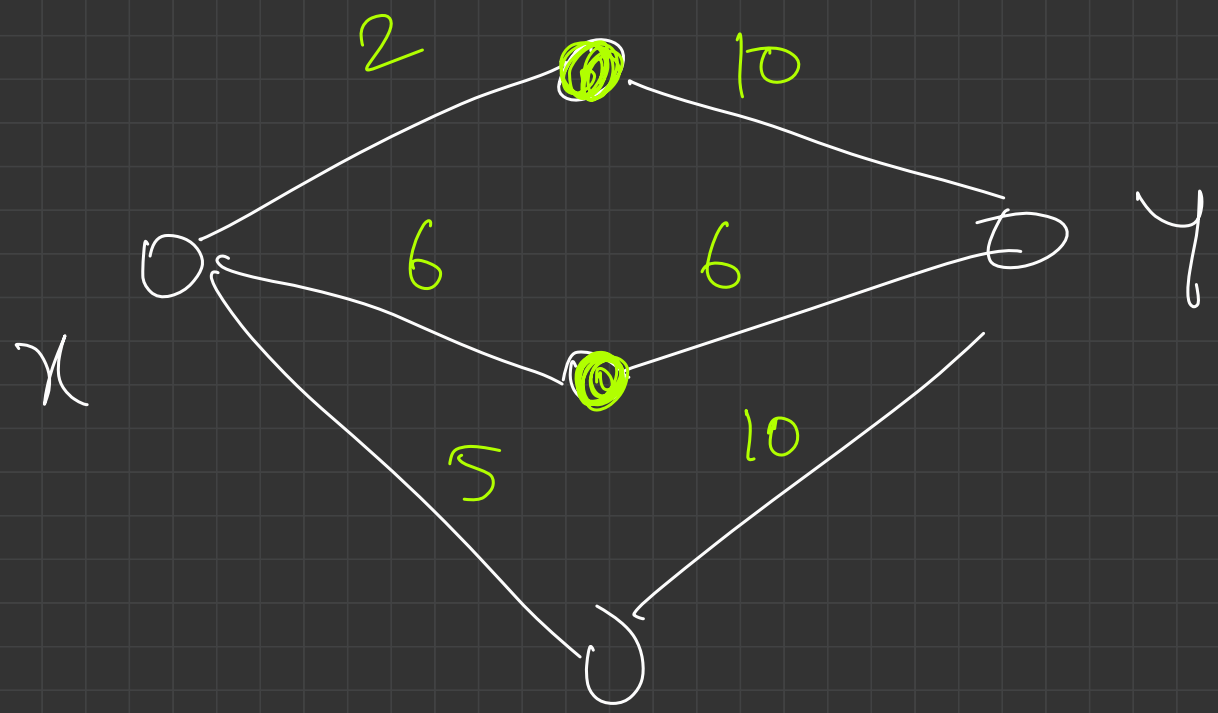
$visit(3) = true$

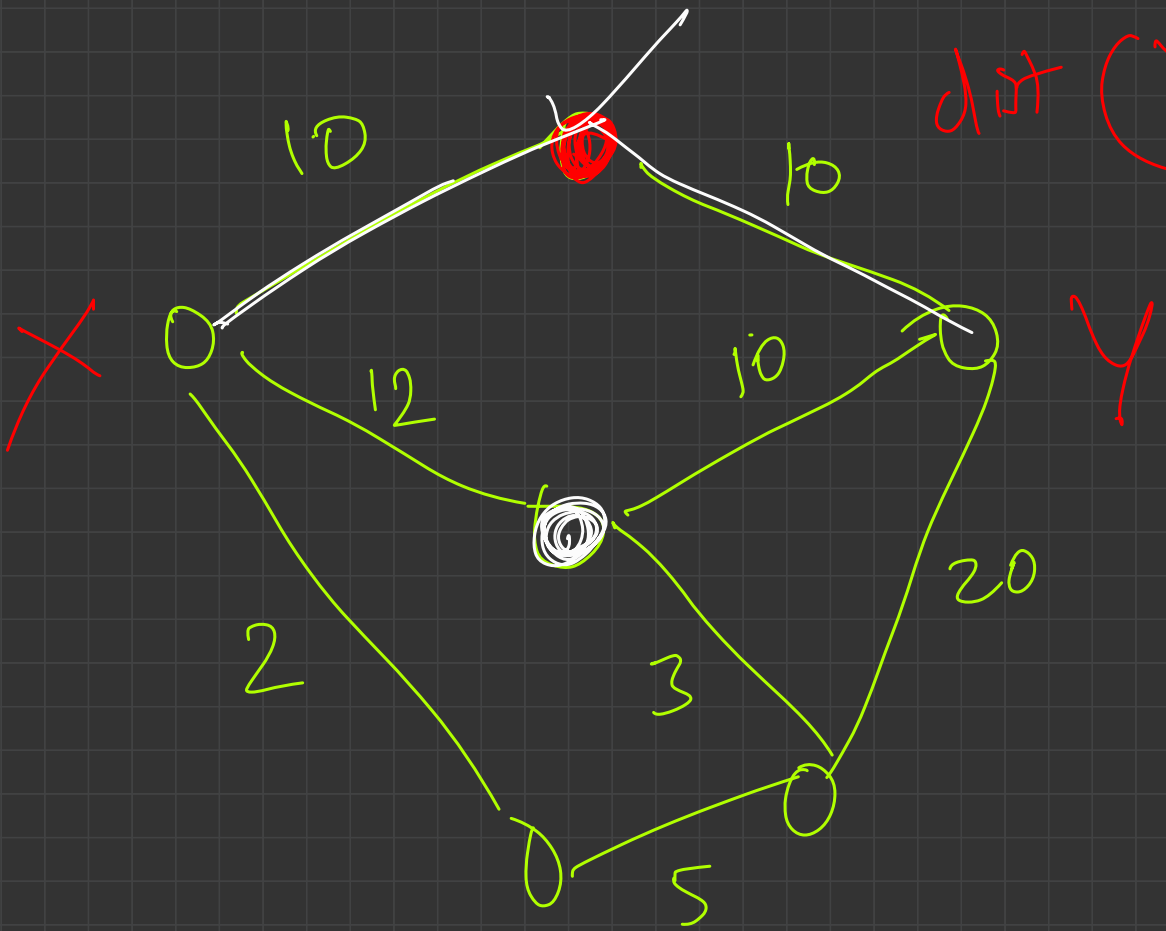


① Given a weighted undirected connected Graph find out the shortest path from source X to Y

→ Dijkstra $\boxed{n \le 10^5 \quad m \le 10^5}$

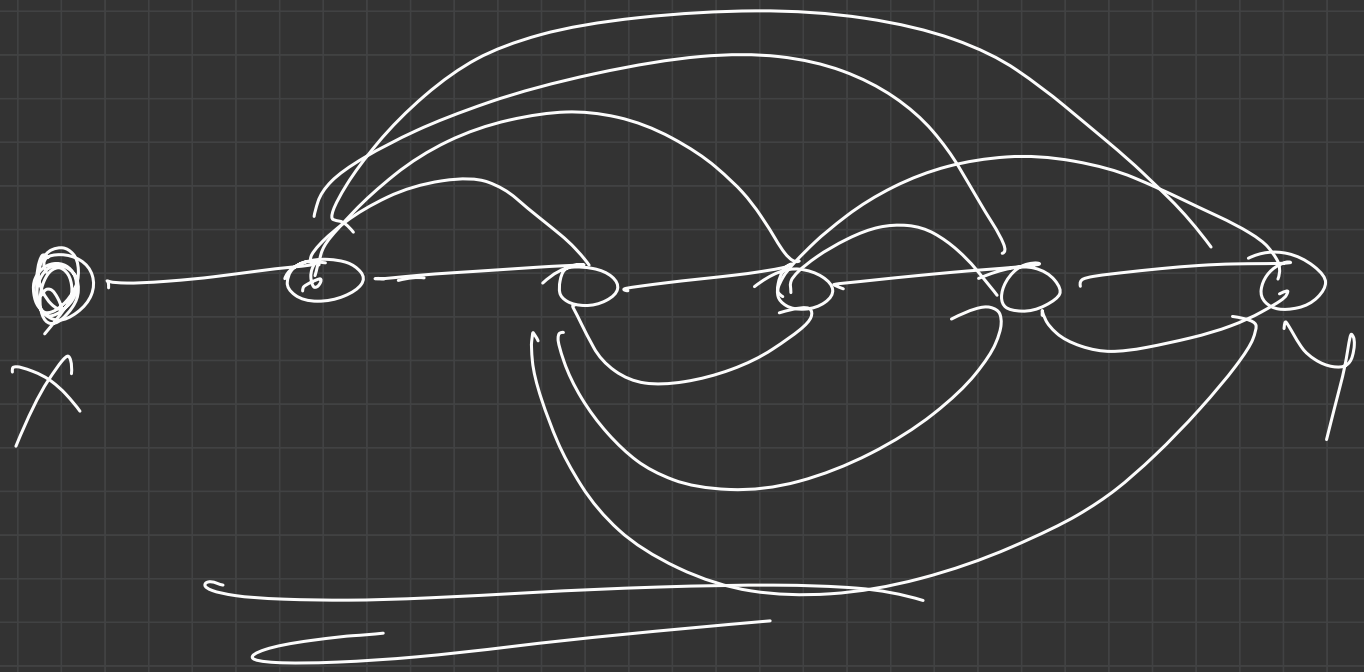
1.1) Consider all shortest paths from X to Y and find out all the nodes that can be on one such path


2
10
6
6
5
10
x
y

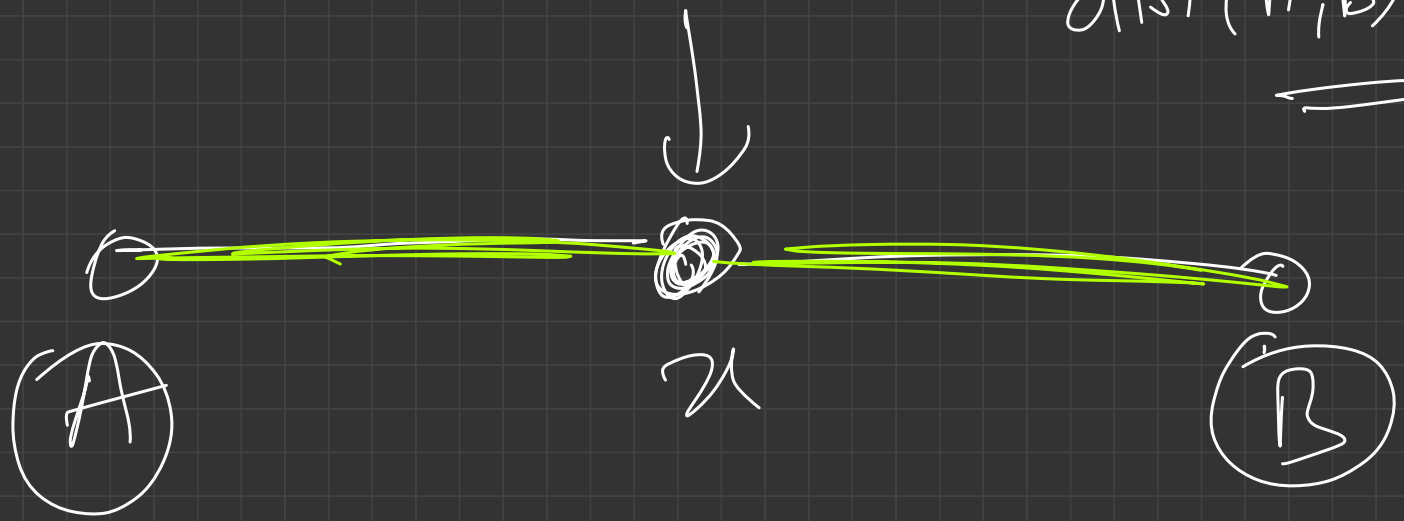


$dist(X, Y) = 20$

$O(n)$


X
Y

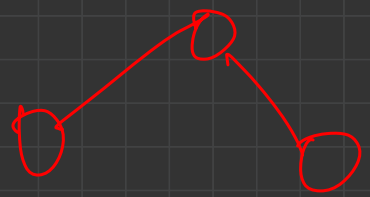
$\text{dist}(A, B) = 15$



$$\text{dist}(A, x) + \text{dist}(x, B) = 15$$

$$= \text{dist}(A, B)$$

① 1 dijkstra from A



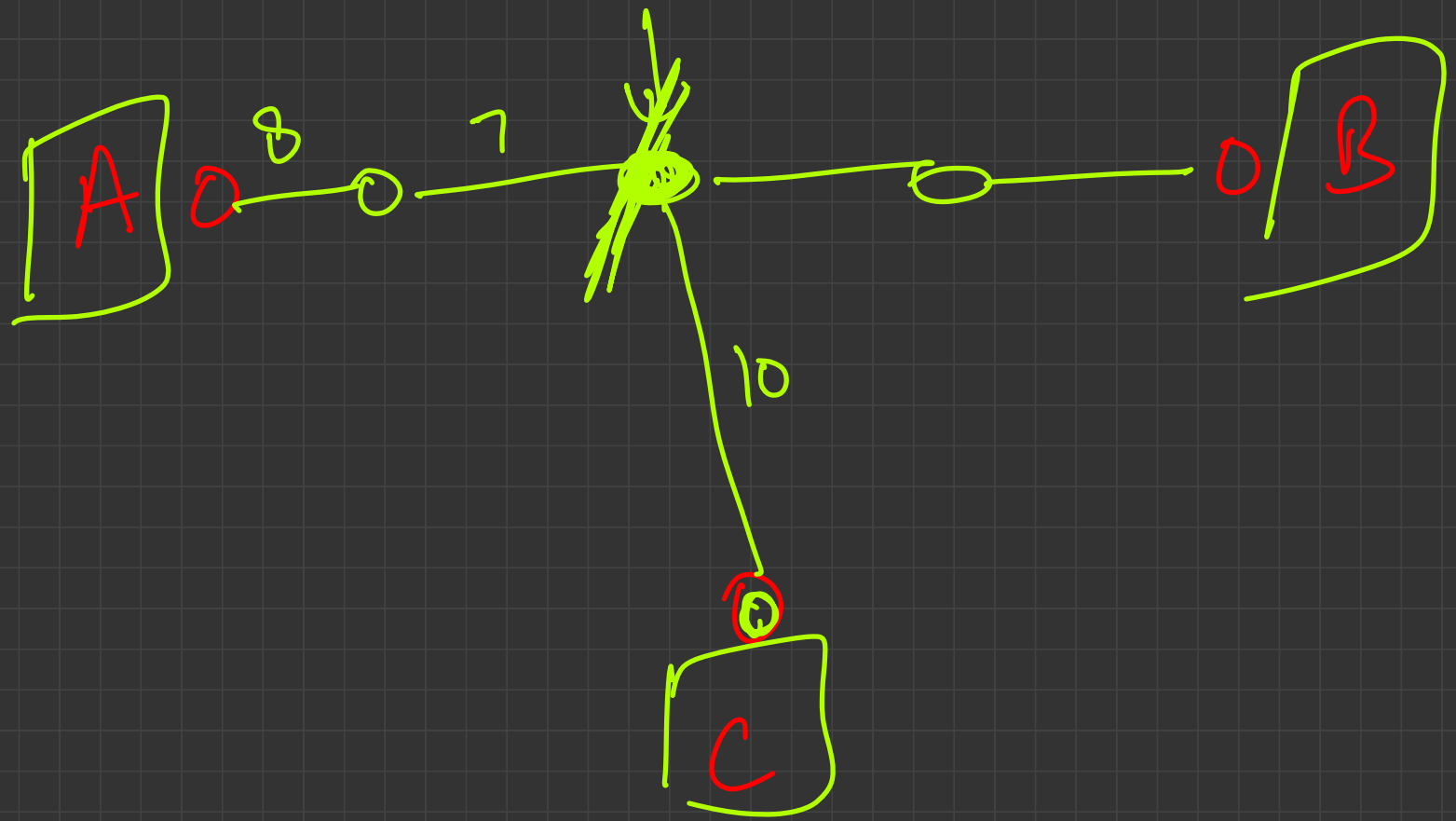
distA = [0]

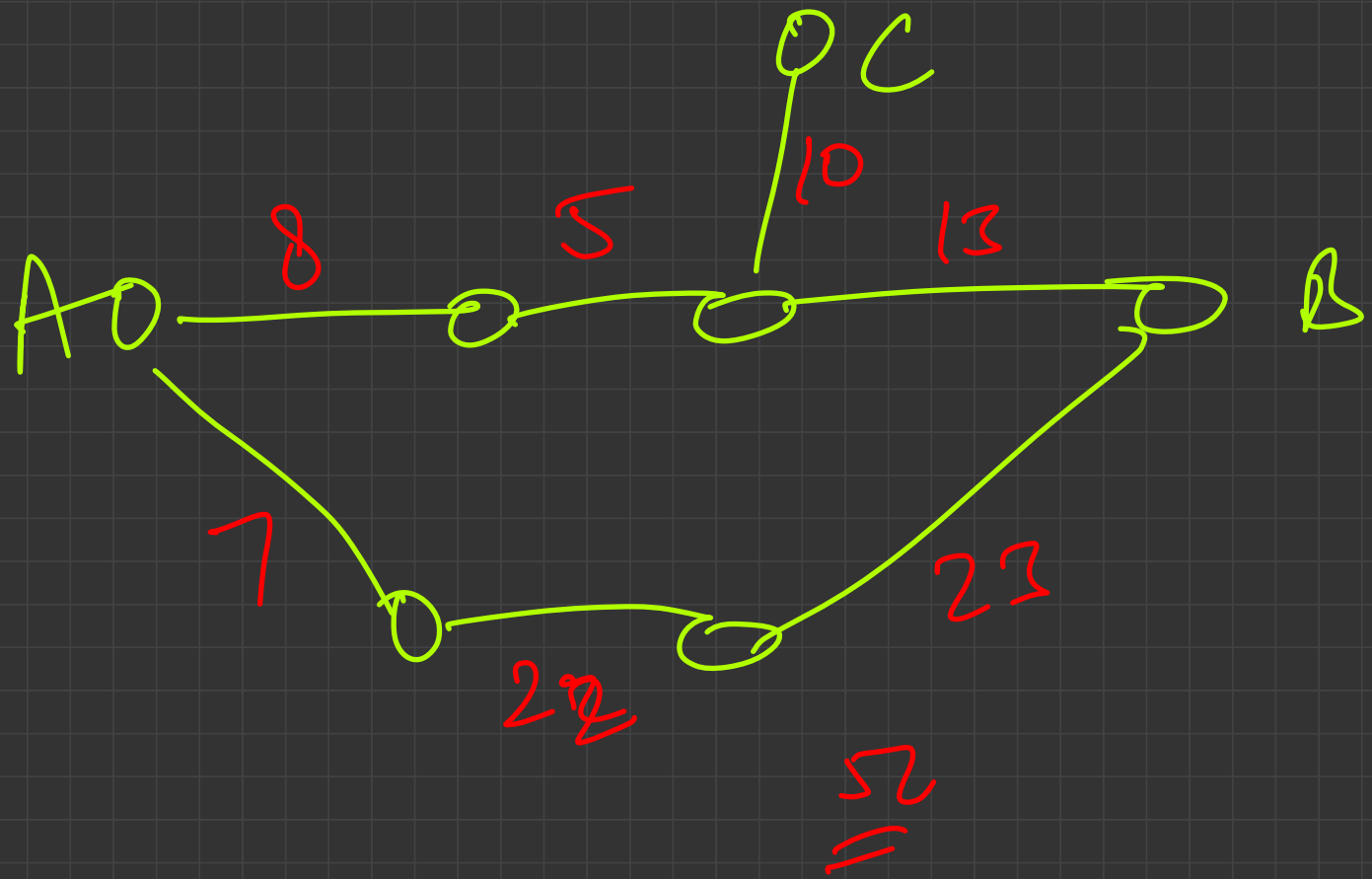
x

② 1 dijkstra from B

distB = [ ]

distA[x] + distB(x) = dist(A, B)


A
8
7
B
10
C


C
10
8
5
13
A
B
7
23
22
52

① Run dijkstra from A

② Run dijkstra from C

if dist C(x) < dist A(x)

remove x from the graph

③ Keep final dijkstra from A

Bellman Ford

- Single Source Shortest Path Algorithm - Idea + Intuition
- Negative edge weights (in directed) without negative cycles
- Proof/Intuition:
    - Principle of Mathematical Induction
- Code
- Breaking Early
- Retrieving the shortest path?
- Finding a negative cycle

Bellman Ford Visualization