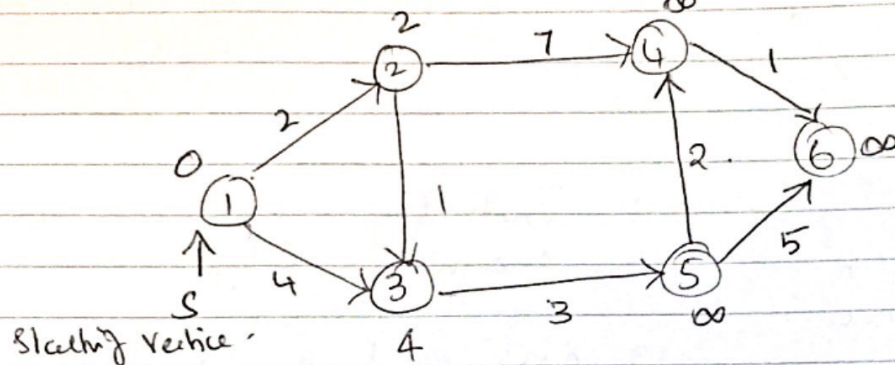


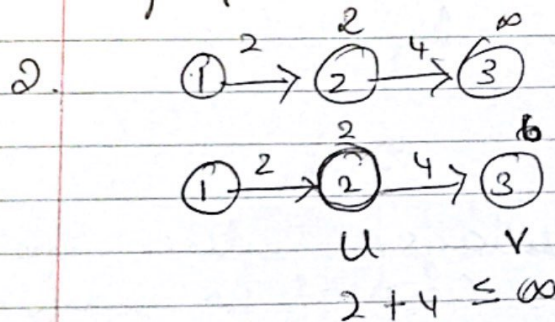
Using min heap

Kruskal's algo  $O(n \log n)$

Dijkstra Algorithm = Single Source Shortest Path



1. D.- algo can be used directed / non-directed graph

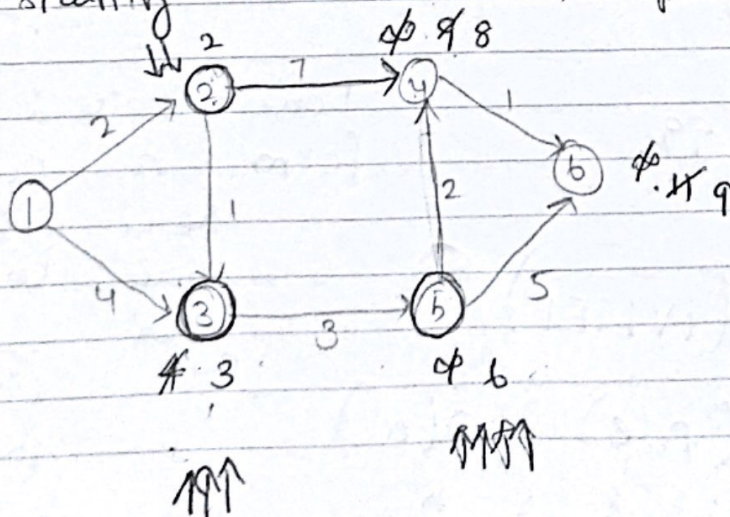


Relaxation

$$\text{if } (d(u) + c(u, v) < d(v))$$

$$d(v) = d(u) + c(u, v)$$

if there is no direct path from dist. starting vertex then put  $\infty$  as ~~dist~~





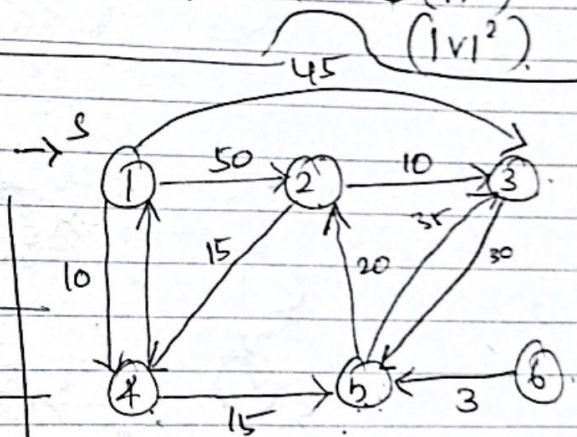
V	dist
2	2
3	3
4	8
5	6
6	9

$|V| = n$  finding shortest path  
 it relaxes, at most it will  
 relax  $|V|$   
 it means  $n \times n = n^2$

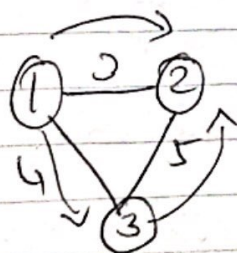
Worst case time of Dijkstra  $O(n^2)$   
 $(|V|^2)$

Ex: Starting Vertex ①

selected vertices	2	3	4	5	6
4	50	45	10	$\infty$	$\infty$
5	50	45	10	25	$\infty$
2	45	45	10	25	$\infty$
3	45	45	10	25	$\infty$
6	45	45	10	25	$\infty$



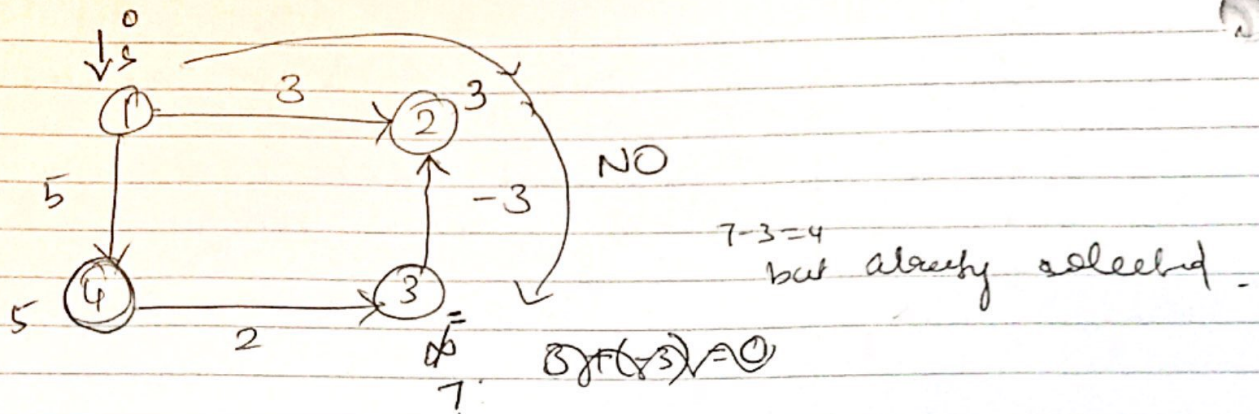
Path toward 5 6 is  $\infty$  only.



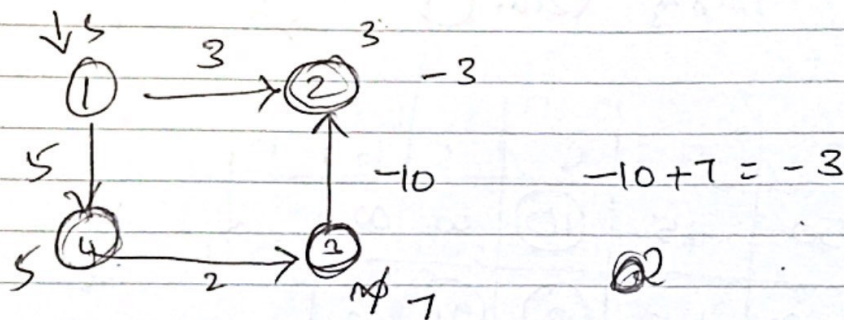
- non directed. if you are not  
 can do this algo then give  
 direction and do this mini  
 path



Ex



-3 negative weight edge.



if we have - edges then dijkstra may or may not work. as already selected we don't check again

in the above case. ② already selected and its dis 3 but from ③ it give -3

so dijkstra did not give shortest path.