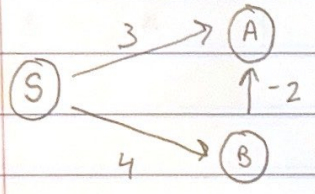
Bellman-Ford Algorithm Theory

Shortest path from one node to all other nodes

Bellman-Ford works on negative edge weights, while Dijikstras does not. (No negative cycle though)

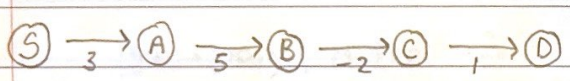
Dijikstra = greedy Algorithm

Bellman-Ford ≠ greedy



Shortest with Dijikstras from (S) to (A), will be 3.

But, with Bellman-Ford we can do (S) to (B), then (B) to (A) for a total of 2 (4-2 =2).



# of vertices = 5

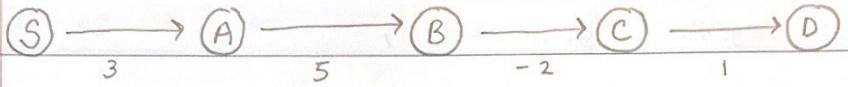
$|V| - 1 = 4$

At most, $|V| - 1$ edges in one of our paths.

$|V|$ = # of vertices

|V| or more edges in a path → repeated vertex → cycle

Simple path = no repeated edges



# of vertices = 5

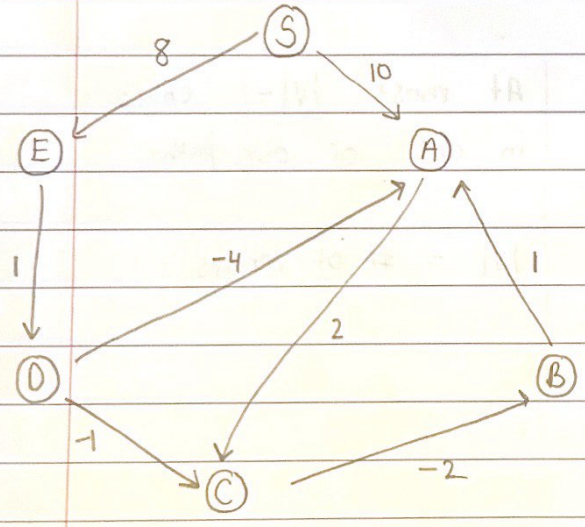
|V| - 1 = 4

Time Complexity : O(|V| · |E|)

|V| = # of vertices

|E| = # of edges

Bellman-Ford Algorithm Example



6 vertices, so we need to do 5 iterations

1st Iteration:

Start:

0	∞	∞	∞	∞	∞
S	A	B	C	D	E

0	10	∞	∞	∞	8
S	A	B	C	D	E

(A → C: 10 + 2)

0	10	∞	12	∞	8
S	A	B	C	D	E

S → A → C → B (12 − 2)

0	10	10	12	9	8
S	A	B	C	D	E

2nd Iteration

Start:

0	10	10	12	9	8
S	A	B	C	D	E

~~(B → E → D → A)~~ (S → E → D → C)

0	5	10	8	9	8
S	A	B	C	D	E

3rd:

0	5	10	8	9	8
S	A	B	C	D	E

0	5	5	7	9	8
S	A	B	C	D	E

We continue you these iterations for #4 & #5 they don't nessarily keep having to change. But, since there is no change from iteration #3 to #4, We stop after #4 to make the algorithm more efficent.

Time complexity: $O(|V| \cdot |E|)$