



NETWORK FLOW

Prepared by,

Chetan Shetty
Assistant Professor
Department of CSE
MSRIT
Bangalore

SHORTEST PATH IN THE GRAPH

- Given a graph G with weights, as described above, decide if G has a negative cycle—that is, a directed cycle C such that

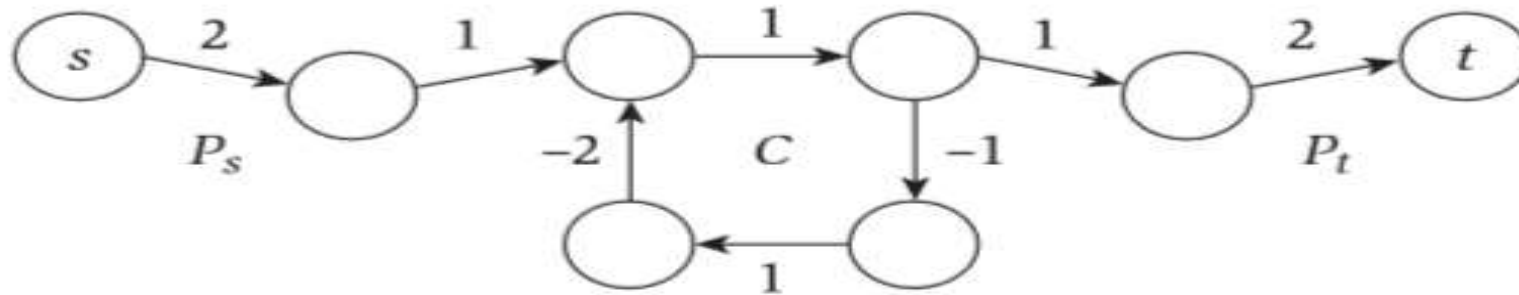
$$\sum_{ij \in C} c_{ij} < 0.$$

- If the graph has no negative cycles, find a path P from an origin node s to a destination node t with minimum total cost:

$$\sum_{ij \in P} c_{ij}$$

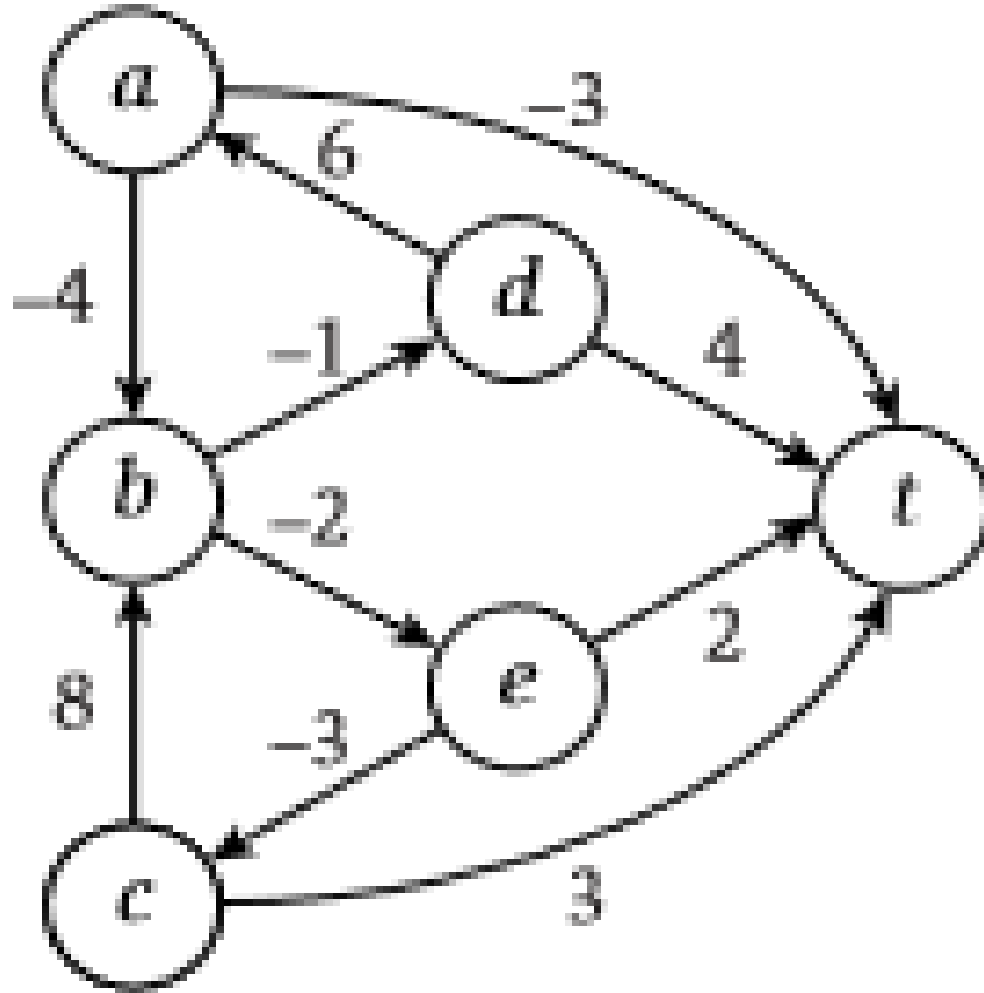
should be as small as possible for any s - t path. This is generally called both the *Minimum-Cost Path Problem* and the *Shortest-Path Problem*.

- Negative cycle corresponds to profitable sequence of transaction. Buy from i_1 , sell it to i_2 , buy from i_2 and sell it to i_3 and so on arriving back to i_1 with a net profit.
- we can build an s - t path of arbitrarily negative cost: we first use P_s to get to the negative cycle C , then we go around C as many times as we want, and then we use P_t to get from C to the destination t .



- We will use dynamic programming to solve the problem of finding shortest path from s to t , when there are negative edge costs but no negative cycles.

Bellman Ford



- $M(i,v) = \min(M[i-1,v], (\min_{w \in V} [M[i-1,w] + C_{vw}]))$

- $M(2,a) = \min(M[1,a], \min([1,t]+3, [1,b]+4))$
 $(-3, 0-3) = -3$
 $(-3, \infty-4) = -3$

- $M(2,b) = \min(M[1,b], \min([1,d]+1, [1,e]+2))$
 $(\infty, 4-1) = 3$
 $(\infty, 2-2) = 0$

- $M(2,c) = \min(M[1,c], \min([1,t]+3, [1,b]+8))$
 $(3, 0+8) = 3$
 $(3, 0+3) = 3$

- $M(2,d) = \min(M[1,d], \min([1,t]+4, [1,a]+6))$
 $(4, -3+6) = 3$
 $(4, 0+4) = 4$

- $\text{Min}(2,e) = \min(2, 3-3) = 0$
 $\min(2, 0+2) = 2$

- $\text{Min}(3,a) = \min(-3, 0-4) = -4$
 $\min(-3, 0-3) = -3$

- $\text{Min}(3,b) = \min(0, 3-1) = 2$
 $\min(0, 0-2) = -2$

- $\text{Min}(3,c) = \min(3, 0+8) = 3$
 $\min(3, 0+3) = 3$

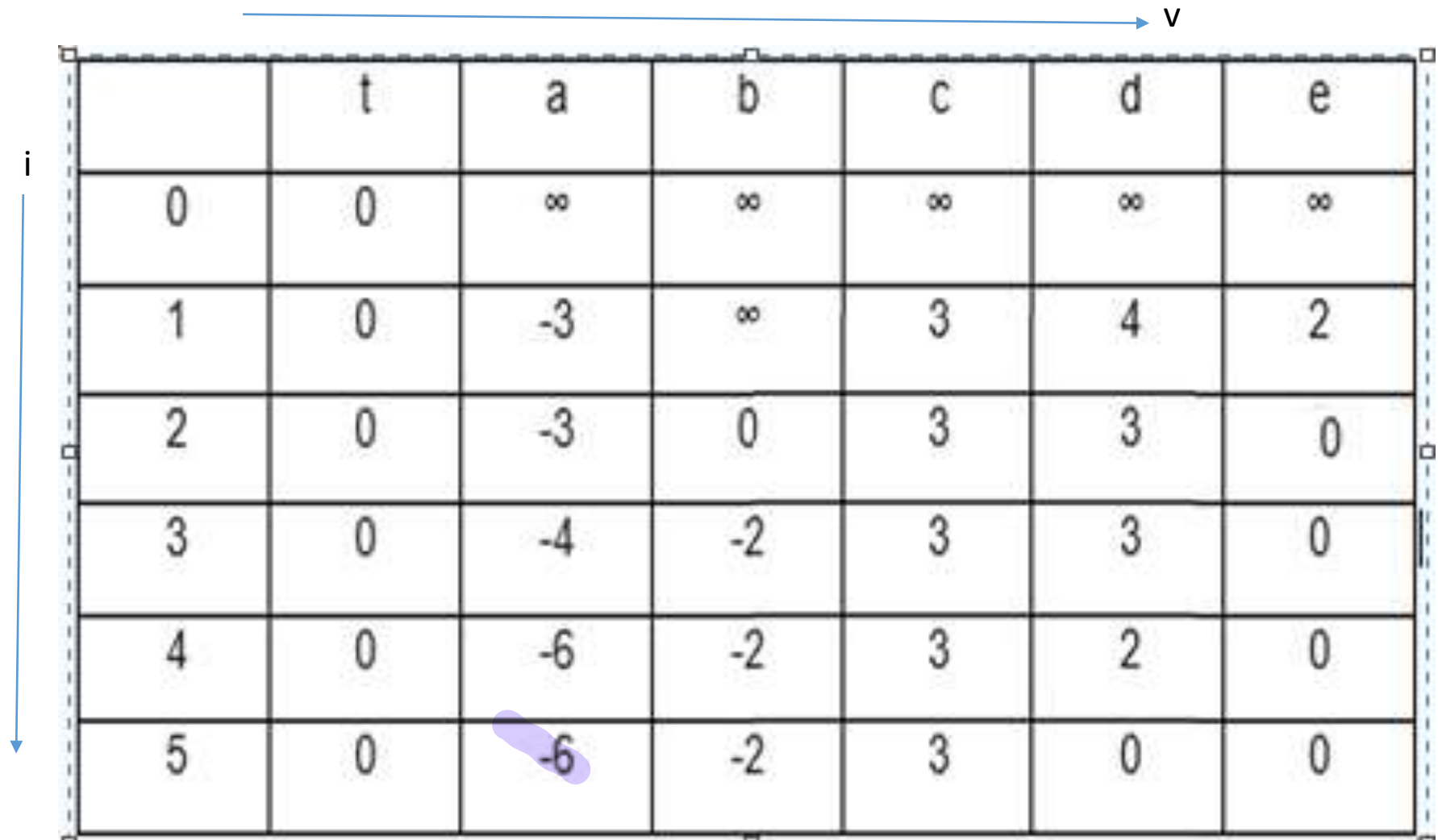
- $\text{Min}(3,d) = \min(3, -3+6) = 3$
 $\min(3, 0+4) = 4$

- $\text{Min}(3,e) = \min(0, 0+2) = 0$
 $\min(0, 3-3) = 2$

- $\text{Min}(4,a) = \min(-4, -2-4) = -6$
 $\min(-4, 0-4) = -4$
- $\text{Min}(4,b) = \min(-2, 3-1) = -2$
 $\min(-2, 0-2) = -2$
- $\text{Min}(4,c) = \min(3, -2+8) = 3$
 $\min(3, 0+3) = 3$
- $\text{Min}(4,d) = \min(3, -4+6) = 2$
 $\min(3, 0+4) = 4$
- $\text{Min}(4,e) = \min(0, 0+2) = 0$
 $\min(0, 3+2) = 0$

- $\text{Min}(5,a) = \min(-6, -2-4) = -6$
 $\min(-6, 0-3) = -3$
- $\text{Min}(5,b) = \min(-2, 2-1) = -2$
 $\min(-2, 0-2) = -2$
- $\text{Min}(5,c) = \min(3, 0+3) = 3$
 $\min(3, -2+8) = 3$
- $\text{Min}(5,d) = \min(2, -6+6) = 2$
 $\min(2, 0+4) = 2$
- $\text{Min}(5,e) = \min(0, 3-3) = 0$
 $\min(0, 0+2) = 0$

If the 6th loop also gets the same result then there is no weighted loop in the graph.



	t	a	b	c	d	e
0	0	∞	∞	∞	∞	∞
1	0	-3	∞	3	4	2
2	0	-3	0	3	3	0
3	0	-4	-2	3	3	0
4	0	-6	-2	3	2	0
5	0	-6	-2	3	0	0

O(VE)

Shortest-Path(G, s, t)

n = number of nodes in G

Array $M[0 \dots n-1, V]$

Define $M[0, t] = 0$ and $M[0, v] = \infty$ for all other $v \in V$

For $i = 1, \dots, n-1$

For $v \in V$ in any order

Compute $M[i, v]$ using the recurrence (6.23)

Endfor

Endfor

Return $M[n-1, s]$

O(V)

O(E)

O(1)

(6.23) If $i > 0$ then

$$\text{OPT}(i, v) = \min(\text{OPT}(i-1, v), \min_{w \in V} (\text{OPT}(i-1, w) + c_{vw})).$$

