2/15

Is 9251629211 a prime number?

(a) Yes (b) NO

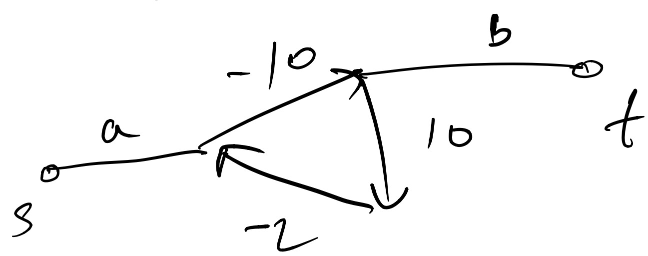
Shortest paths in graphs

Input: $G=(V,E)$ is a directed graph.

Edge $e=(i,j)$ has a cost c_{ij}.

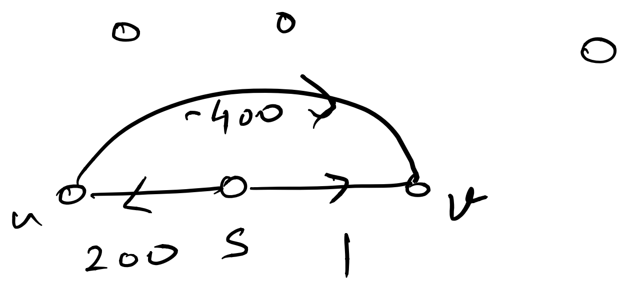
(c_{ij}'s need not be positive)

Assume: G has no negative weight cycles.



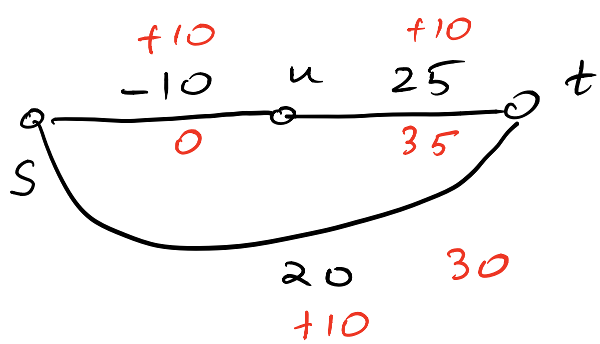
Greedy approaches?

→ Try Dijkstra's algorithm?



→ [Baxter's suggestion]

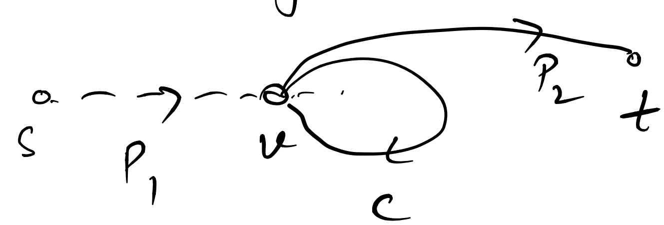
Add a large number C to each edge weight.



Dynamic programming approaches?

Obs If G has no negative cycles, there is a shortest path from s to t which is **Simple** (no repeated vertices).

Pf: If a shortest path P is not simple, $\exists$ a cycle in P.



$$P = P_1 \circ C \circ P_2$$

$$\text{Cost}(P) = \text{cost}(P_1) + \text{cost}(C) + \text{Cost}(P_2)$$

So P': $P_1 \circ P_2$

$$\text{Cost}(P') = \text{Cost}(P_1) + \text{Cost}(P_2) \le \text{Cost}(P),$$

since no negative cycles.

$\Rightarrow \text{Cost}(P') = \text{Cost}(P)$. $\square$

Task: $s \to t$ (shortest path)

Floyd-Warshall Algorithm

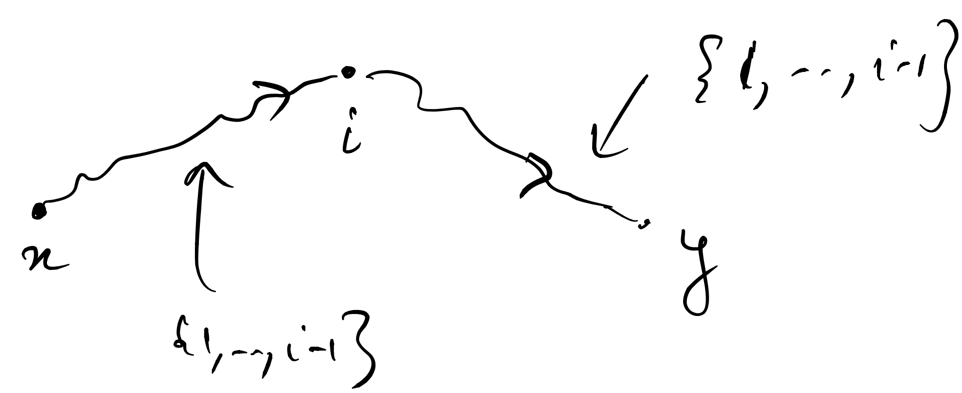
$OPT(x, y, i)$: Shortest path from x to y using nodes $\{1, 2, \dots, i\}$. $(x, y \in V)$

think of $V = \{1, \dots, n\}$

$OPT(x, y, i)$: (1) All intermediate nodes in the shortest path is in $\{1, \dots, i-1\}$.

$OPT(x, y, i) = OPT(x, y, i-1)$.

(2)



$OPT(x, y, i) = OPT(x, i, i-1) + OPT(i, y, i-1)$

$\circledast$

$$OPT(x, y, i) = \min \left\{ \begin{array}{l} OPT(x, y, i-1), \\ OPT(x, i, i-1) + OPT(i, y, i-1) \end{array} \right\}$$

→ Initialize $OPT(x, y, 0) = c_{xy}$

→ for $i = 1$ to n

  for $x \in V$

    for $y \in V$

      Compute $OPT(x, y, i)$ using $\circledast$

  end for

end for

Return $OPT(s, t, n)$.

Running time: $O(n^3)$

Bellman - Ford algorithm.

$OPT(v, i) \Rightarrow$ length of shortest path from s to v using at most i edges.

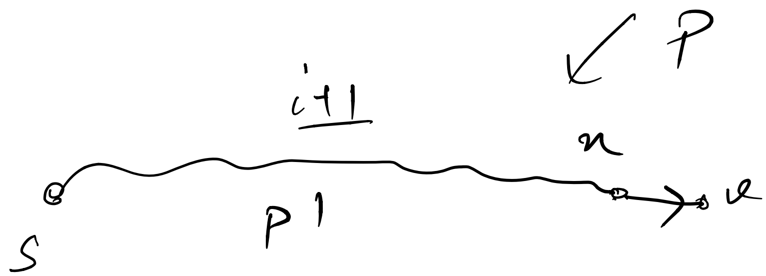
We want to compute $OPT(t, n-1)$.

$OPT(v, i+1)$:

(1) Shortest path uses exactly $i+1$ edges

(2) strictly less than $(i+1)$ edges :

$$OPT(v, i+1) = OPT(v, i)$$



$$\text{cost}(P) = \text{cost}(P') + C_{xv}$$

length of P' is i.

$$OPT(v, i+1) = OPT(x, i) + C_{xv}$$

(+) $$OPT(v, i+1) = \min \left\{ OPT(v, i), \min_{x \in V} \left\{ OPT(x, i) + C_{xv} \right\} \right\}$$

→ Initialize $OPT(x, 0) = 0$ if $x = s$, ∞ o\w

→ for $i = 1$ to n
  for $v \in V$

  (+) to compute $OPT(v, i)$.

and for.

end for;

Output $OPT(t, n-1)$.

Running time: n^2 iterations,

Each iteration: $O(n)$ time.

$O(n^3)$ [see book to improve running time to $O(mn)$].