Math 409: Discrete Optimization

Today: Directed graphs and shortest paths

Directed graphs

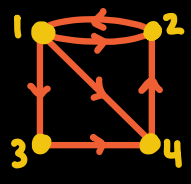
directed graph (or **di-graph**): $G=(V,E)$

V = "vertices"

E = "edges" or "arcs" have the form (i,j) where $i,j \in V$

ordered pair (as opposed to unordered $\{i,j\}$)

Ex: $G=(\{1,2,3,4\}, \{(1,2),(2,1),(1,3),(1,4),(3,4),(4,2)\})$



A **directed path** is a digraph on distinct vertices $\{v_0,..,v_k\}$ with edge set $P=\{(v_0,v_1),(v_1,v_2),..,(v_{k-1},v_k)\}$

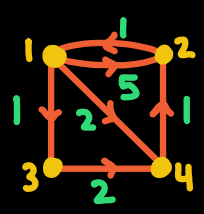
↳ called a v_0-v_k path

Shortest Path Problem

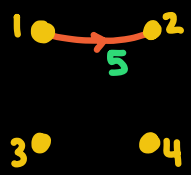
Input: directed graph $G=(V,E)$,

edge costs $c: E \to \mathbb{R}$, vertices $s,t \in V$

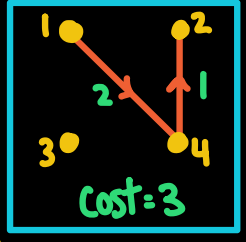
Goal: Find an s-t path $P \subseteq E$ minimizing $C(P)=\sum_{e \in P} c(e)$

Ex 1: 

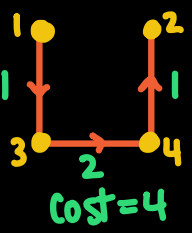
s=1
t=2



cost = 5

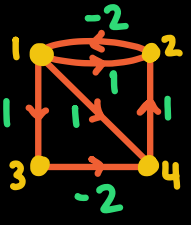


cost = 3

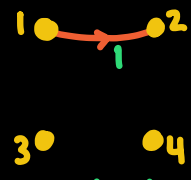


cost = 4

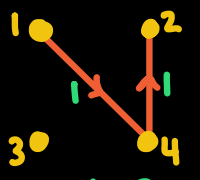
also min cost walk from s→t (allowing repeated vertices)

Ex 2: 

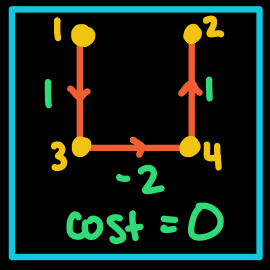
S=1
t=2



cost = 1



cost = 2



cost = 0

Note: there are walks s→t of arbitrarily low cost!

Alternate a=(1,2), b=(2,1): cost(aba)=0 cost(ababa)=-1 cost(abababa)=-2

Two types of assumption we'll make:

(1) $c(e) \geq 0$ for all $e \in E$ Note: (1) ⇒ (2)

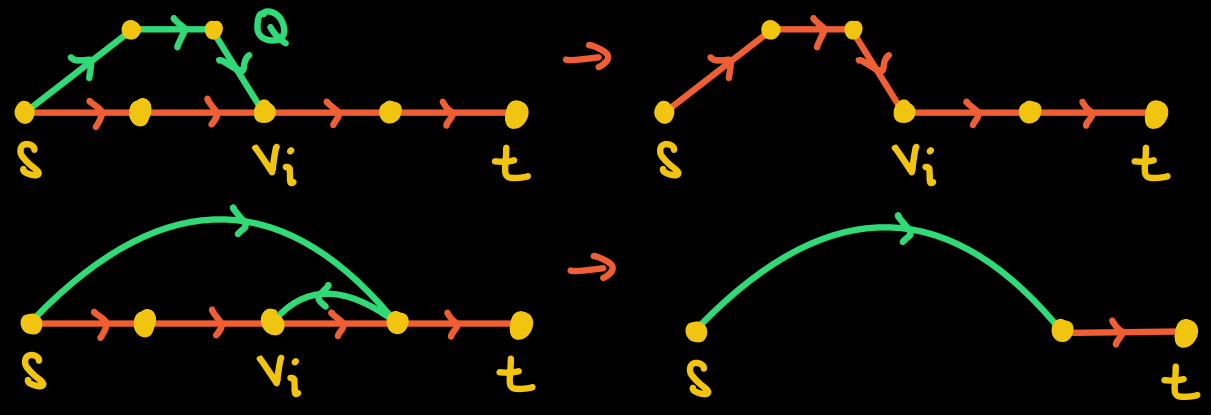
(2) All cycles have cost ≥ 0

We'll need to assume one of these in order to get a polynomial time algorithm. (See notes for details)

For now we take (1) $c(e) \geq 0$ for all $e \in E$.

Bellman's principle: If $P = \{(s, v_1), (v_1, v_2), \ldots, (v_k, t)\}$ is the shortest s-t path, then for every $i \leq k$ $\{(s, v_1), (v_1, v_2), \ldots, (v_{i-1}, v_i)\}$ is the shortest s-v_i path.

(Idea) If Q is a shorter s-v_i path, use it to make an s-t path shorter than P



Take $\ell = \max\{j : Q \text{ visits } v_j\}$ and append s-v_ℓ path in Q to v_ℓ-t path in P.

Idea: to compute shortest s-t path, iteratively build up best s-v paths for all v.

DIJKSTRA'S ALGORITHM

Input: $G=(V,E)$ digraph, $s \in V$, costs $c: E \to \mathbb{R}_{\geq 0}$

Output: min cost $\ell(v)$ of s-v path for all $v \in V$

(1) Set $\ell(s)=0$, $R=\{s\}$

For $w \in V \setminus \{s\}$, set $\ell(w) = \begin{cases} c(s,w) & \text{if } (s,w) \in E \\ \infty & \text{if } (s,w) \notin E \end{cases}$

(2) While $R \neq V$ do

(3) select $v \in V \setminus R$ attaining $\min\{\ell(v) : v \in V \setminus R\}$

(4) For $w \in V \setminus R$ with $(v,w) \in E$, do $\ell(w) := \min\{\ell(w), \ell(v)+c(v,w)\}$

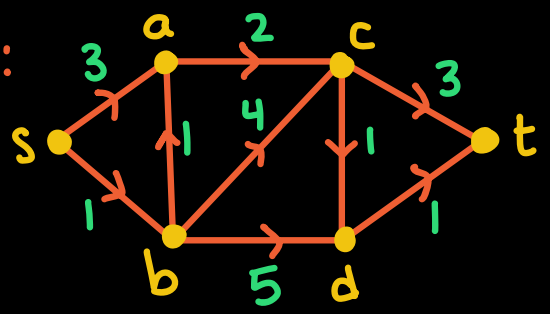
(5) Set $R = R \cup \{v\}$

Proof of correctness next time!

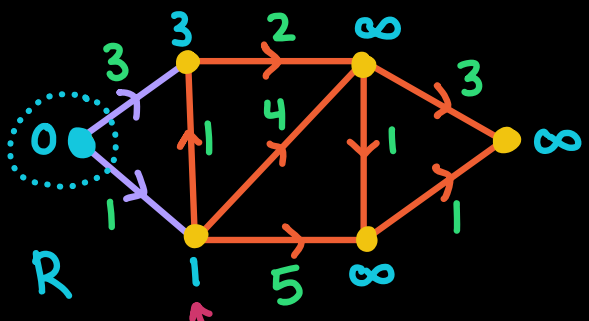
Note: to compute shortest path (not just its length) store shortest s-v path whenever $\ell(v) \neq \infty$

If $\ell(v) + c(v,w) < \ell(w)$ in (4), update s-w path to stored s-v path with (v,w) appended

Ex:

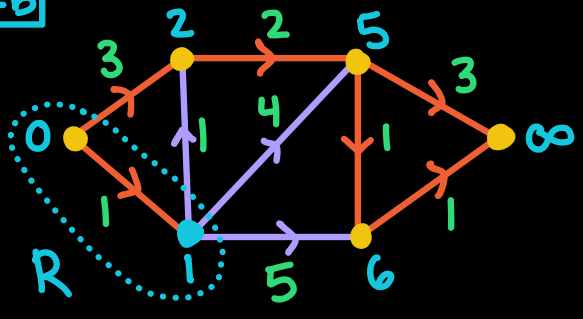


v	R	$\ell(s)$	$\ell(a)$	$\ell(b)$	$\ell(c)$	$\ell(d)$	$\ell(t)$
–	{s}	0	3	1	∞	∞	∞
b	{s,b}	0	2	1	5	6	∞
a	{s,a,b}	0	2	1	4	6	∞
c	{s,a,b,c}	0	2	1	4	5	7
d	{s,a,b,c,d}	0	2	1	4	5	6



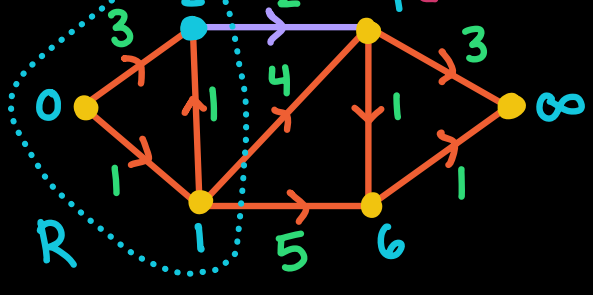
min val not in R

v=b

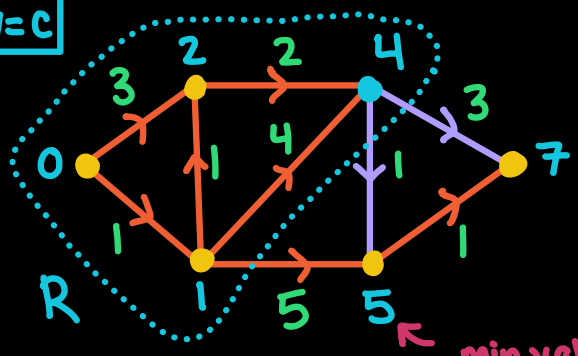


v=a

min val not in R

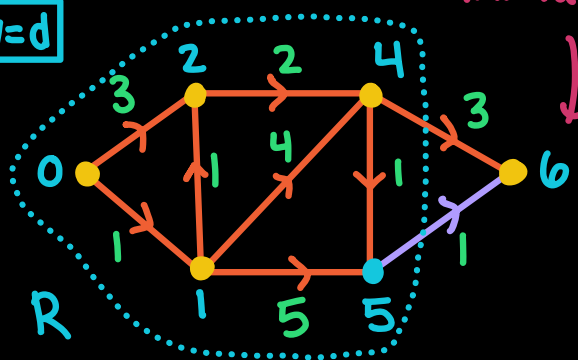


v=c



min val ∉ R

v=d



v=t