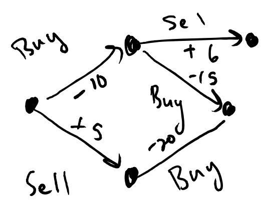
S. KIMMEL

Dijkstra

Good: v. fast! $O(m \log n)$ run time

Bad: • Not good if have distributed graph like internet because need to maintain global heap
• Not good if have negative weights

e.g. financial transaction graph:

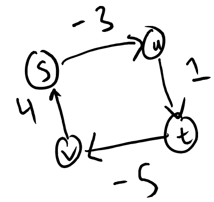


New Approach: Dynamic Programming - Bellman Ford

(actually used for internet routing!)

Problem: What to do when G has negative weight cycle?

def: Cycle is a path from $v \in V$ back to $v \in V$, and doesn't repeat other vertices



A) Return $-\infty$

B) Should return shortest cycle free path

↳ NP-complete problem. Best know algorithm is exponential in n, m

Bellman Ford:

Input: directed graph $G = (V, E)$, edge costs l_e, vertex $s \in V$

Output: Negative cycle in G

or (if no negative cycle)

Shortest paths from s to all other $v \in V$

For now, assume no neg. wt. cycle in G (but neg. wt. edges OK!)

Q: If a graph G has no negative weight cycles, what is an upper bound on the number of edges in a shortest path?

A) no bound B) m C) n D) n-1

Proof:
- For contradiction, suppose shortest path has > n-1 edges.
- Then must visit same vertex twice ⇒ cycle
- All cycles have non-negative weight, so if remove, get shorter path, a contradiction!

S.KIMMEL

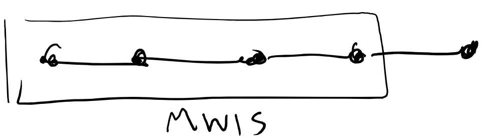
To Create D.P. (dynamic programming) algorithm:

$\boxed{1.}$ Think of form of optimal solution.

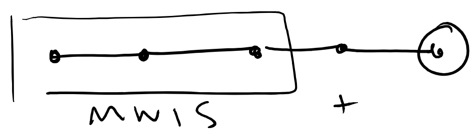
- WMIS on line: (ii) $v_n \in S$ or (i) $n \notin S_n$

$\boxed{2.}$ How do you write in terms of optimal solution to smaller problem?

(i)

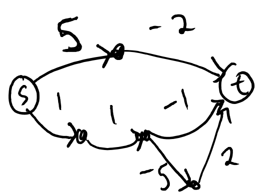


(ii)



Problem: on general graphs, hard to order subproblems

Q:



What is shortest path from s to t with at most 2 edges? at most 3 edges?

A) 3, 1

B) 2, 0

C) 3, -1

D) 2, 1

We'll use max # of edges in path to order our subproblems

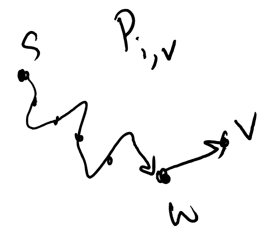
S.KIMMEL

Let $P_{i,v}$ = shortest $s-v$ path with at most i edges (∞ if no $s-v$ path) (assume unique $\forall\ v, i$)

Q: Fill in blue

<u>Case 1</u>: If $P_{i,v}$ has $\leq (i-1)$ edges,

$P_{i,v} = P_{i-1,v}$



<u>Case 2</u>: If $P_{i,v}$ has i edges, then

$$P_{i,v} = P_{i-1,w} + (w,v)$$

for some $w : (w,v) \in E$

Pf: <u>Case 1</u>: • $\ell(P_{i,v}) \leq \ell(P_{i-1,v})$ since extra edge can only help

• If $\ell(P_{i,v}) < \ell(P_{i-1,v})$, then there is a shorter path than $P_{i-1,v}$ with $\leq (i-1)$ edges, a contradiction

$\Rightarrow \ell(P_{i,v}) = \ell(P_{i-1,v})$

by uniqueness, $P_{i,v} = P_{i-1,v}$

<u>Case 2</u>: Suppose $P_{i,v} = p + (w,v)$ where p is a path from s to w, $p \neq P_{i-1,v}$

• p longer $\Longrightarrow$ $P_{i,v}$ is not optimal

• p can't be less than $P_{i-1,v}$ by def

$\Longrightarrow P_{i,v} = P_{i-1,w} + (w,v)$ for some w.

S.KIMMEL

Q: How many subproblems must be evaluated to calculate $P_{i,v}$?

A) $n+1$ B) n C) $1+|\{u:(u,v)\in E\}|$ D) $|\{u:(u,v)\in E\}|$

Case 1: 1 → $P_{i-1,v}$

Case 2: $P_{i-1,w}$, for each $w \in \{u:(u,v)\in E\}$

* Cycles permitted b/c i keeps from infinite cycling around

$\boxed{\text{3 (Dynamic Programming)}}$ Create recurrence relation

Let $L_{i,v}$ be length of path $P_{i,v}$ (∞ if $\circ$ path)

Q. Base Case: $L_{0,s} = 0$ $L_{0,v} = \infty \ \forall \ v \in V-s$

Recurrence: $L_{i,v} = \min \begin{cases} L_{i-1,v} \\ \min\limits_{(w,v)\in E} L_{i-1,w} + c_{(w,v)} \end{cases}$

Correctness: Using proof on previous page, $P_{i,v}$ must be related to one of $1+|\{w:(w,v)\in E\}|$ subproblems. We look at all (exhaustive search)