Dijkstra

Good: very fast! O(m log n) run time

Bad: not good if have distributed graph like internet
   - Not good if have negative weights

E.g. financial transaction graph:

- Buy ➔ Sell ➔ Buy ➔ Sell

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 ∈ V back to v ∈ V, and doesn't repeat other vertices

A) Return -∞
B) Should return shortest cycle free path

⇒ NP-complete problem. Best known algorithm is exponential in nm

Bellman Ford:

Input: directed graph $G = (V, E)$, edge costs $c_e$, vertex $s ∈ V$

Output: Negative cycle in $G$
   or (if no negative cycle)
   Shortest paths from $s$ to all other $v ∈ V$

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 a path has $>n-1$ edges.

- Then must visit same vertex twice $\Rightarrow$ cycle
- All cycles have non-negative weight, so if remove, get shorter path, a contradiction!
To Create D. P. (dynamic programming) algorithm:

1. Think of form of optimal solution.
   - WMIS on line: \( v_n \in S \) or \( n \notin S \)

2. How do you write in terms of optimal solution to smaller problems?
   - (i)
   - (ii)

Problem: on general graphs, hard to order subproblems

Q: What is shortest path from 5 to 4 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
Let \( P_{i, v} = \) shortest \( s-v \) path with at most \( i \) edges 
(\( \infty \) if no \( s-v \) path) \hspace{1cm} (\text{assume unique} \ A \ v, i)

**Case 1:** If \( P_{i, v} \) has \( \leq (i-1) \) edges,
\[
P_{i, v} = P_{i-1, v}
\]

**Case 2:** If \( P_{i, v} \) has \( i \) edges, then
\[
P_{i, v} = P_{i-1, w} + (w, v)
\]
for some \( w: (w, v) \in E \)

**Proof:**

**Case 1:** \( \ell(P_{i, v}) \leq \ell(P_{i-1, v}) \) 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, \( P_{i, v} = P_{i-1, v} \)

**Case 2:** Suppose \( P_{i, v} = P + (w, v) \) where \( p \) is a path from \( s \) to \( w \), \( p \not\supset P_{i-1, v} \)

- \( p \) longer \( \Rightarrow P_{i, v} \) is not optimal.
- \( p \) can't be less than \( P_{i-1, v} \) by def

\[
\Rightarrow P_{i, v} = P_{i-1, w} + (w, v) \text{ for some } w.
\]
Q: How many subproblems must be evaluated to calculate $P_{i,v}$?
A) $n+1$  B) $n$  C) $1 + \left| \{ u : (u,v) \in E \} \right| + D \left| \{ u : (u,v) \in E \} \right|

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

$\mathcal{O}$ Cycles permitted b/c $i$ keeps from infinite cycling around

3 (Dynamic Programming) Create recurrence relation

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

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

Recurrence: $L_{i,v} = \min \left\{ L_{i-1,v} \right\}$

$\min_{(w,v) \in E} \left( L_{i-1,w} + c_{(w,v)} \right)$

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