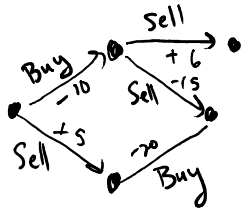
S. KIMMEL

Dijkstra

Good: Fast! $O(m \log n)$ run time

Bad:
- Need to maintain global heap (impossible for internet)
- Fails with negative weights (see HW)

Financial Transactions:



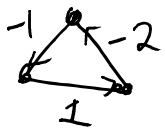
Bellman-Ford

(actually used for internet routing!)

- Dynamic Programming
- Slower but no global heap, negative weights ok.

Bellman Ford:

Input: directed graph $G=(V,E)$, edge weights ℓ_e, vertex $s \in V$, assume no negative cycles



← Neg. cycle

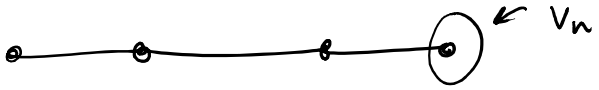
Output: Shortest paths from s to all other $v \in V$

S. KIMMEL

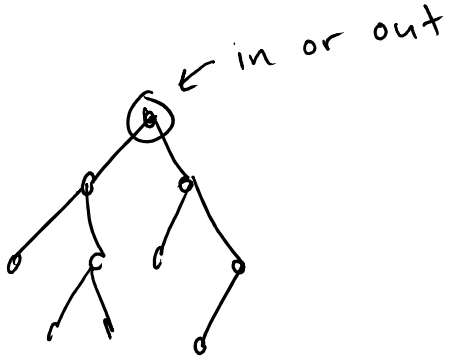
To Create D.P. (dynamic programming) algorithm:

1. Think of options for one (final) piece of optimal solution
2. Write optimal solution in terms of optimal solution to subproblem:

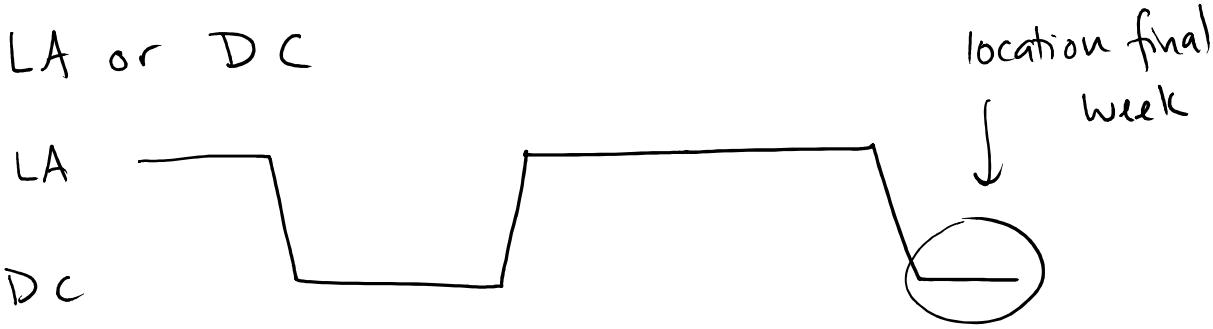
- WMIS on line: $v_n \in S$ (ii) or $v_n \notin S_n$ (i)



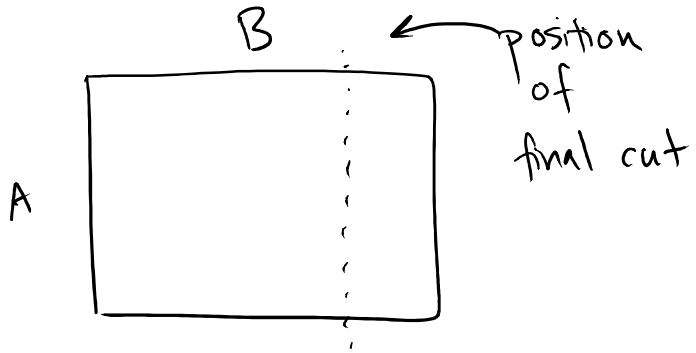
- WMIS on tree



- LA or DC



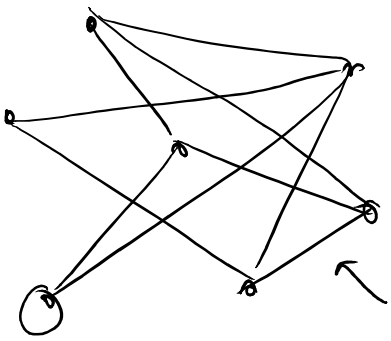
-



3. Create recurrence relation for objective function
4. Fill in array. START from base case! Go in opposite direction of recurrence
5. Work backwards to find optimal solution if desired.

S. KIMMEL

Problem: on general graphs, hard to define subproblems

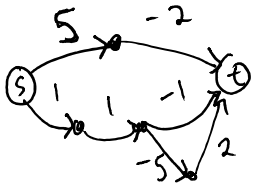


Idea: Edge present or not?

- Why this edge?
- Solution could change wildly if edge present
- Not simple recursive relationship

Same for removing vertex

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

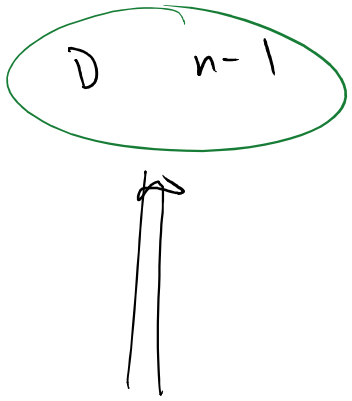
D) 2, 1

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

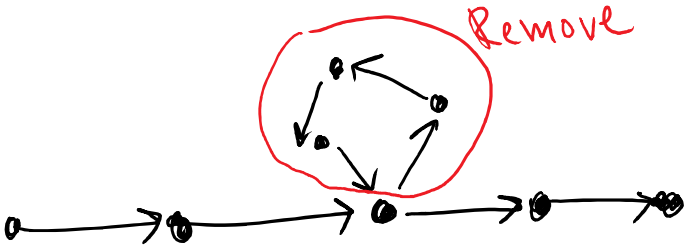
S.KIMMEL

Q: If a graph G has no negative weight cycles, how many edges are in the shortest path. (Let $|V|=n$, $|E|=m$

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



Proof:
- For contradiction, suppose the shortest path has more than n-1 edges.
- Then path must visit same vertex twice, so it contains a cycle
- All cycles have non-negative weight, so if remove cycle from path, the result is a shorter path, a contradiction.



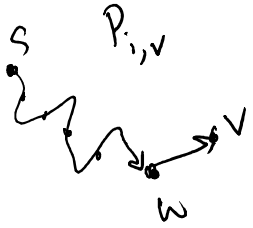
S. KIMMEL

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

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



To prove, use proof by contradiction:

- Case 1: Suppose for contradiction $P_{i,v}$ has less than i edges, but $P_{i,v} = P^* \neq P_{i-1,v}$.
Then P^* is a path with at most $i-1$ edges from s to v. But $P_{i-1,v}$ is the shortest such path, so $\ell(P_{i-1,v}) \leq \ell(P^*)$, so you can always choose $P_{i,v} = P_{i-1,v}$ to get shortest possible path

Case 2: Assume for contradiction $P_{i,v}$ has i edges, but is $P^*_w + (w,v)$ for $P^*_w \neq P_{i-1,w}$.

...

(rest is similar to Case 1)

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: $P_{i-1,v}$ (1 subproblem)

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

* Cycles are allowed – limit on i keeps from infinite cycles

3 (Dynamic Programming) Create recurrence relation

Let $L_{i,v}$ be length of path $P_{i,v}$ (∞ if no path)

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

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

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)

S. KIMMEL

Q:

Pseudo Code :

Note: Start at Base case!

$L[i, s] = 0 \quad \forall \; i$

$L[0, v] = \infty \quad \forall \; v \in V - s$

for $(i = 1$ to $n-1)\{$ ← max # edges required is n-1 if no neg. cycles (see HW)

for $(v \in V - s)$

$$L[i, v] = \min \{ L[i-1, v], \min_{(w,v) \in E} L[i-1, w] + l(w, v) \}$$

- Assume have inverse adjacency list $A'_G[v] = \{ u : (u, v) \in E \}$
- Do for loop over $A'_G[w]$

$\}$

Q: What is runtime of Bellman Ford? (Pick strongest bound)

A) $O(n^2)$ B) $O(mn)$ C) $O(n^3)$ D) $O(m^2)$

$$\sum_{i=1}^{n-1} \sum_{v} \left(1 + \sum_{(w,v) \in E} 1 \right)$$

$$n(n + m) = O(nm)$$

($n \leq m$ if G is connected)