Dynamic Programming

[illegible]

Note Title 2/15/2010

Repeated use of a small (polynomial) number of subproblems to solve a given problem optimally.

We will see:

- shortest paths (Dijkstra)
- shortest paths using at most k edges
- all pairs shortest paths

- longest increasing subsequence
- edit distance / longest common substring

- Knapsack
- Matrix chains
- polygon triangulation
- max ind. set in a tree

$G = (V, E) \quad l : E \to \mathbb{R}_+$

$l(e) \geq 0$

Find shortest path from s to t in G.

Shortest s-t path = shortest s-u path + $l(u,t)$ for some $u \in V \setminus \{t\}$.

Maintain tentative distances from s to all nodes.

$T(s) = 0 \quad T(u) = l(s,u)$ if $(s,u) \in E$

$T(u) = \infty \quad (s,u) \notin E$

$S = \{s\}$. Find node u with minimum $T(u)$

- and add to S.
- Update $T(v) = \min\{T(v), T(u) + (u,v)\}$

Nodes are included in S in order of increasing distance from s.

Complexity? $O(|V|^2)$ - $|V|$ steps
- each is a minimum

Can be made $O(|E| + |V| \log |V|)$ with efficient data structures

Shortest s-t path with at most k edges.

$P(a, b, j)$: length of shortest a, b path with at most j edges.

$$P(a,b,j) = \min\left\{ \min_{c} P(a,c,j-1) + \ell(c,b),\ P(a,b,j-1) \right\}$$

Fix a.

	u_1	u_2	u_i
1	$\ell(a,u_1)$	..	$\ell(a,u_i)$
2			
⋮			
k			

Fill this table.

Time : Size of Table × time per entry
$\leq$ $k \cdot n \cdot n$

All pairs shortest paths.

Using Dijkstra : $|V|\,|E|$.

Negative edges allowed (but no negative cycles)

Dijkstra does not work.

Floyd-Warshall.

Order nodes $1, 2, \ldots n$

$d(i,j,k)$ = shortest ij path using intermediate nodes numbered k or less.

$$d(i,j,k) = \min\{d(i,j,k-1),\ d(i,k,k-1) + d(k,j,k-1)\}$$

<u>Table</u> is $n \times n \times n$

Time $= O(n^3) = O(|V|^3)$.

longest increasing subsequence

$s_1, s_2, \quad \dots \quad, s_n$

$S(j)$ = longest inc. subseq. ending with s_j.

$$S(j) = \max \left\{ 1 + \max_{\substack{i < j \\ s_i < s_j}} S(i) \right\}$$

Return $\max_j S(j)$.

To find sequence also store $prev(j)$.

$\parallel$ to shortest paths.

edit distance

RAIN –   RAIN
– WIND   WIND

3   4

consider edit distances between all pairs of prefixes of the strings

$D(i,j) = \text{Edit distance}(a_1 \ldots a_i, b_1 \ldots b_j)$

possible alignments

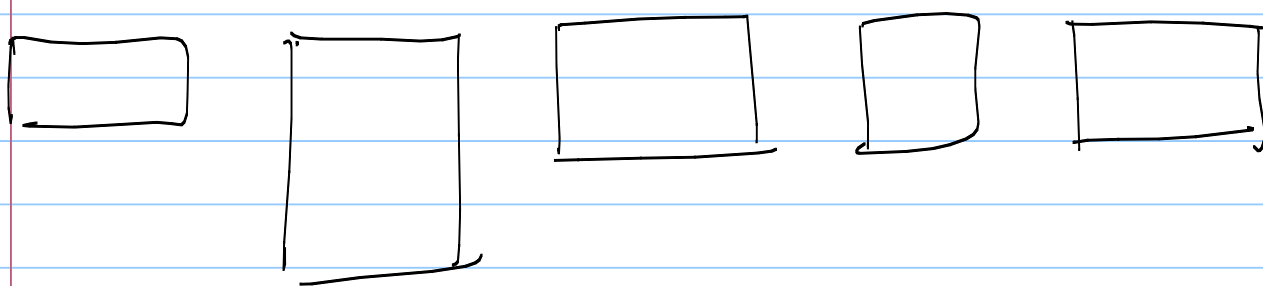
—	a_i	a_i
b_j	—	b_j

cost = 1   1   1 if $a_i \neq b_j$
                0 if $a_i = b_j$

$$D(i,j) = \text{Min}\{1 + D(i,j-1),\ 1 + D(i-1,j),\ \chi(a_i \neq b_j) + D(i-1,j-1)\}$$

Matrix chain

$A_1 \times A_2 \quad \ldots \quad A_n$



$m_1 \times m_2 \qquad m_2 \times m_3 \qquad m_3 \times m_4$

Can do $\quad m_1 m_2 m_3 + m_1 m_3 m_4 = m_1 m_3 (m_2 + m_4)$

OR $\quad m_2 m_3 m_4 + m_1 m_2 m_4 = m_2 m_4 (m_1 + m_3)$

$2 \times 6 \qquad 6 \times 10 \qquad 10 \times 6 \qquad 20 \cdot 12$

or $\quad 36 \cdot 12$

What is the best sequence?

$C(i,j)$ = Best sequence of mult. for $A_i \cdot A_{i+1} \cdots A_j$

$$C(i,j) = \min_{i \le k \le j} \{ C(i,k) + C(k,j) + m_i m_k m_k \}$$

Max Independent in a tree.

root r.

Subtree at Node u. $T(u)$

if $u \in I$ then children of $u \notin I$.

Let $I(u)$ = max ind set in $T(u)$

$$I(u) = \max \Big\{ 1 + \sum_{v:\text{ grandchildren of } u} I(v) \;,\; \sum_{u:\text{ children of } u} I(u) \Big\}$$