

CS1510, Design and Analysis of Algorithms, Fall 2006

Patchrawat Uthaisombut

## Dynamic Programming

---

### general coin change problem

Let  $C(v)$  be the minimum number of coins of denominations 4, 3, 1 needed to make change for  $v$  dollars

$$C(v) = \min \{ C(v-4) + 1, C(v-3) + 1, C(v-1) + 1 \}$$

### knapsack problem

Let  $P(i, w)$  be the maximum profit that obtainable, with weight at most  $w$ , from items in the set  $\{1, 2, \dots, i\}$

$$P(i, w) = \max \{ P(i-1, w), P(i-1, w-w_i) + p_i \}$$

### chained matrix multiplication

Let  $M(i, j)$  be the minimum number of multiplications needed to compute  $A_i A_{i+1} \dots A_j$

$$M(i, j) = \min_{i \leq k \leq j-1} \{ M(i, k) + M(k+1, j) + p_{i-1} p_k p_j \}$$

### optimal binary search tree

Let  $C(i, j)$  be the cost of an optimal binary search tree containing real keys  $i, \dots, j$  and dummy keys  $i-1, \dots, j$

$$C(i, j) = \min_{i \leq r \leq j} \{ C(i, r-1) + c(r+1, j) + w(i, j) \}$$

### longest common subsequence

Let  $L(i, j)$  be the length of a longest common subsequence of  $x_1 x_2 \dots x_i$  and  $y_1 y_2 \dots y_j$

$$L(i, j) = \begin{cases} L(i-1, j-1) + 1 & \text{if } x_i = y_j \\ \max \{ L(i, j-1), L(i-1, j) \} & \text{if } x_i \neq y_j \end{cases}$$

### sequence alignment

Let  $C(i, j)$  be the cost of an optimal alignment of  $x_1 x_2 \dots x_i$  and  $y_1 y_2 \dots y_j$

$$C(i, j) = \max \left\{ \begin{array}{l} L(i-1, j-1) + \alpha_{x_i y_j} \\ L(i, j-1) + \delta \\ L(i-1, j) + \delta \end{array} \right\}$$

### Floyd's shortest path algorithm

Let  $L(i, j, k)$  be the length of a shortest path from  $i$  to  $j$  where the internal nodes are from the set  $\{1, 2, \dots, k\}$

We want  $L(i, j, n)$  for all  $i, j$

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

### bitonic euclidean TSP

Label the vertices using positive integer from left to right

For  $i < j$ , let  $L(i, j)$  be the length of a shortest bitonic spanning path for vertices in the set  $\{1, 2, \dots, i\}$

where one path ends at vertex  $i$  and the other path ends at vertex  $j$

We want  $\min_{1 \leq j \leq i-1} \{ L(n, j) + d_{j,n} \}$

$$L(i, j) = \min \{ L(i-1, j) + d_{i-1,i}, \min_{1 \leq k \leq i-2} L(i-1, k) + d_{k,i} \}$$

### maximum contiguous sublist

Let  $V(i)$  be the value of a maximum contiguous sublist of  $s_1, s_2, \dots, s_i$  ending with  $s_i$

We want  $\max_{1 \leq i \leq n} V(i)$

$$V(i) = \max \{ V(i-1) + s_i, s_i \}$$

### Longest Increasing Subsequence (LIS)

Let  $L(i)$  be the length of a longest increasing subsequence of  $x_1, x_2, \dots, x_i$  ending with  $x_i$

We want  $\max_{1 \leq i \leq n} L(i)$

$$L(i) = \max_j \{ L(j) + 1 \} \text{ where the maximum is over all } j \text{ such that } 0 \leq j < i \text{ and } x_j < x_i$$

### making change with limited number of coins

Let  $C(i, v)$  be the size of a smallest multiset that is a subset of  $\{d_1, d_2, \dots, d_i\}$  to make change for  $v$  dollars

Let  $C(i, v)$  be  $\infty$  if there is no such set

$$C(i, v) = \min \{ C(i-1, v), C(i-1, v-d_i) + 1 \}$$

### price-collecting path

Let  $S(i, j)$  be the largest weight of any shortest path from node  $(0, 0)$  to node  $(i, j)$  in grid graph  $G$

$$S(i, j) = \max \{ S(i, j-1), S(i-1, j) \} + w_{i,j}$$

### rocket assembly

Let  $C(i, k)$  be the cost of an optimal assembly sequence to assemble pieces  $i, \dots, k$

$$C(i, k) = \min_{i \leq j \leq k-1} (C(i, j) + C(j+1, k) + f(i, j, k))$$

### word segmentation

Let  $C(i)$  be the quality of an optimal segmentation for the string containing letters  $s_1 s_2 \dots s_i$

$$C(i) = \max_{0 \leq j \leq i-1} C(j) + \text{quality}(s_{j+1} s_{j+2} \dots s_i)$$

### printing neatly

Let  $C(i)$  be the penalty of an optimal layout for words  $i, \dots, n$

We want to compute  $C(1)$

$$C(i) = \min_j \left\{ C(j+1) + \left( L - \sum_{k=i}^j w_k \right)^2 \right\}$$

where the minimum is taken over all  $j$  such that  $i \leq j \leq n-1$  and  $\sum_{k=i}^j w_k \leq L$

Initial condition

$$C(i) = 0 \text{ if } \sum_{k=i}^n w_k \leq L$$


---