Dynamic Programming

P. O. D. P. — Subproblems of an optimal solution are themselves optimal

Matrix Chain Product

\[ A_1, A_2, \ldots, A_n \]

To

\[ p_1, \ldots, p_n \]

The Min

\[ (A_i \cdots A_j) \]

\[ C(i, j) = \min_{i \leq h < j} (C(i, h) + C(h+1, j) + p_i \cdot p_{h+1} \cdot p_j) \]

\[ C(i, i) = 0 \]

Catalan Numbers
(exponential)

Repeated subproblems — Memoization — \( O(n^2) \)

Keep record

Expanded Memoization

\[ C[i, j] - best cut \]
\[ K[i, j] - location of first cut \]
Triangulation of Convex Polygons

Let $n$ be the number of vertices of the polygon.

For any integer $k$, $0 < k < n-1$, let $c(i,j)$ be the minimum cost of triangulating the subpolygon with vertices $i, i+1, \ldots, j$.

The recurrence relation is:

$$c(i,j) = \min_{i < k < j} \left( c(i,k) + c(k,j) + \overline{ik} + \overline{kj} \right)$$

The optimal solution is:

$$c(i, i+\text{mod}(n,i)) = 0 \quad \text{for} \quad 0 < i < n$$

The dynamic programming table $c[i,j]$ can be used to store the minimum cost for each subpolygon. The overall time complexity is $\Theta(n^2)$ for the Dynamic Programming solution. The number of operations is $\sum (j-i) \leq \Theta(n^3)$.

The constant $\overline{ik}$ represents the cost of the edge $ik$.
Longest common subsequence

There are $2^m$ subseq of $x_1 \ldots x_m$

There are $2^n$ subseq of $y_1 \ldots y_n$

$2^n \cdot 2^m = \frac{1}{2} \cdot 2^{n+m}$

$x_m = y_n$

$x_m \neq y_n$

$O(1)/\text{memo}$

$n \times m \text{ memo} \Rightarrow O(nm)$
Optimum B.S.T.