Optimal Binary Search tree
Optimal binary search trees

- e.g. binary search trees for 3, 7, 9, 12;

(a) (b) (c) (d)
Optimal binary search trees

- **n** identifiers: $a_1 < a_2 < a_3 < \ldots < a_n$
- $P_i$, $1 \leq i \leq n$: the probability that $a_i$ is searched.
- $Q_i$, $0 \leq i \leq n$: the probability that $x$ is searched where $a_i < x < a_{i+1}$ ($a_0 = -\infty$, $a_{n+1} = \infty$).

$$\sum_{i=1}^{n} P_i + \sum_{i=1}^{n} Q_i = 1$$
Identifiers : 4, 5, 8, 10, 11, 12, 14
Internal node : successful search, $P_i$
External node : unsuccessful search, $Q_i$

The expected cost of a binary tree:

$$\sum_{n=1}^{n} P_i \times \text{level}(a_i) + \sum_{n=0}^{n} Q_i \times (\text{level}(E_i) - 1)$$

The level of the root : 1
The dynamic programming approach

Let \( C(i, j) \) denote the cost of an optimal binary search tree containing \( a_i, \ldots, a_j \).

The cost of the optimal binary search tree with \( a_k \) as its root:

\[
C(1, n) = \min_{1 \leq k \leq n} \left\{ P_k + \left[ Q_0 + \sum_{i=1}^{k-1} (P_i + Q_i) + C(1, k - 1) \right] + \left[ Q_k + \sum_{i=k+1}^{n} (P_i + Q_i) + C(k+1, n) \right] \right\}
\]
General formula

\[
C(i, j) = \min_{i \leq k \leq j} \left\{ P_k + \left[ Q_{i-1} + \sum_{m=i}^{k-1} (P_m + Q_m) + C(i, k-1) \right] + \left[ Q_k + \sum_{m=k+1}^{j} (P_m + Q_m) + C(k+1, j) \right] \right\}
\]

\[
= \min_{i \leq k \leq j} \left\{ C(i, k-1) + C(k+1, j) + Q_{i-1} + \sum_{m=i}^{j} (P_m + Q_m) \right\}
\]
Computation relationships of subtrees

e.g. n=4

Time complexity : $O(n^3)$
when $j-i=m$, there are $(n-m)$ C(i, j)'s to compute.
Each C(i, j) with $j-i=m$ can be computed in $O(m)$ time.

$O(\sum_{1 \leq m \leq n} m(n-m)) = O(n^3)$