# *UNIT-04. DYNAMIC PROGRAMMING*

1. All pairs shortest paths
2. Single source shortest paths (Bellman Ford Algorithm)
3. Optimal Binary search trees
4. 0/1 Knapsack problem
5. LCS problem
6. Travelling sallsperson problem.

* what is dynamic programming?

Dynamic programming is an algorithm design technique that can be used when the solution to a problem can be viewed as a result of sequence of decisions. DP works on principle of optimality.

* what is principle of optimality?

An optimal sequence of decisions has the property that whatever the initial state and decisions are the remaining decision must constitute an optimal decision sequence with regard to the state resulting from the first decision.

i.e in an optimal sequence of decisions, each subsequence's must also be optimal.

* what are elements of dynamic program:-

1. Optimal substructure
2. Overlapping subproblems
3. Memoization.

1. Optimal Substructure:- A problem has an optimal substructure if an optimal solution can be constructed efficiently from the optimal solutions of its subproblems.

Scanned with OKEN Scanner

i.e an optimal solution to a problem contains within the optimal solution to subproblems optimal solution to the entire problem is build in a bottom up manner from optimal solutions to subproblems.

i.e we can solve larger problems given the solutions to its smaller problems.

2. Overlapping subproblems: The recursive algorithm to a problem is solving same problems multiple times. So the problem has overlapping subproblems.

3. Memoization:- A function remembers values it was called with and the consequent results. Store solution to subproblem in a table, don't recompute.

The solution to each subproblem is stored in a table, to be looked up in the table.

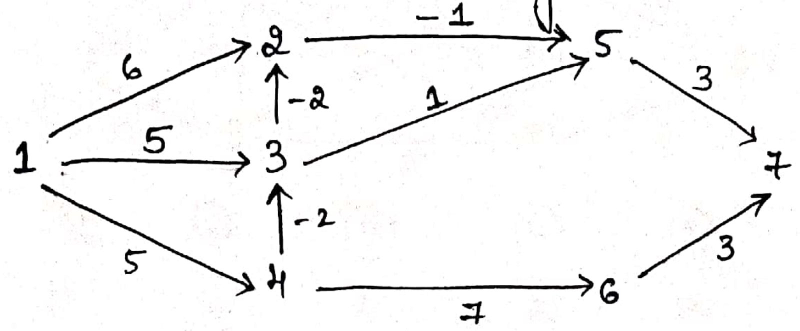
| Greedy Algorithm | Dynamic Programming |
|---|---|
| 1) Uses static approach chooses 1st decision irrespective of remaining decisions. | 1) Uses principle of ~~optimally~~. optimality. |
| 2) Greedy algorithm always run in polynomial time complexity | 2) DP computes cost of all feasible solution (TC may be exponential) |
| 3) It failed to solve some optimisation problem. | 3) DP is used to solve any optimisation problem. |
| 4) Memory usage is low | 4) Memory usage is high. |
| 5) Faster than DP | 5) Slower than greedy approach. |

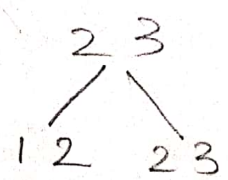
* Bellman Ford Algorithm:-

Bellman Ford algorithm will work even when the graph is having negative weights. It will not work when the graph is having any cycle of negative length.

Scanned with OKEN Scanner

Q: Explain Bellman Ford Algorithm with an example



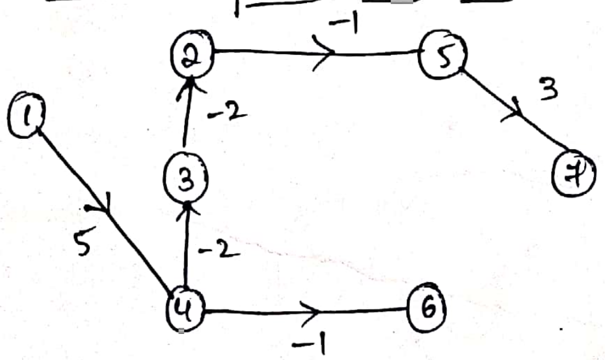




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1→ | 0 | 6 | 5 | 5 | ∞ | ∞ | ∞ |
| 2→ | 0 | 3 | 3 | 5 | 5 | 4 | ∞ |
| 3→ | 0 | 1 | 3 | 5 | 2 | 4 | 7 |
| 4→ | 0 | 1 | 3 | 5 | 0 | 4 | 5 |
| 5→ | 0 | 1 | 3 | 5 | 0 | 4 | 3 |
| 6→ | 0 | 1 | 3 | 5 | 0 | 4 | 3 |
| 7→ | 0 | 1 | 3 | 5 | 0 | 4 | 3 |

| 2 | | 3 | | 4 |
|---|---|---|---|---|
| 1 | 3 | 1 | 4 | 1 |
| 0+6 | 5-2 | 0+5 | 5-2 | 0+5 |
| 0+6 | 3-2 | 0+5 | 5-2 | 0+5 |
| 0+6 | 3-2 | 0+5 | 5-2 | 0+5 |
| 0+6 | 3-2 | 0+5 | 5-2 | 0+5 |
| 0+6 | 3-2 | 0+5 | 5-2 | 0+5 |
| 0+6 | 3-2 | 0+5 | 5-2 | 0+5 |

| 5 | | 6 | 7 | |
|---|---|---|---|---|
| 2 | 3 | 4 | 5 | 6 |
| 6-1 | 5+1 | 5-1 | ∞+3 | ∞+3 |
| 3-1 | 3+1 | 5-1 | 5+3 | 4+3 |
| 1-1 | 3+1 | 5-1 | 2+3 | 4+3 |
| 1-1 | 3+1 | 5-1 | 0+3 | 4+3 |
| 1-1 | 3+1 | 5-1 | 0+3 | 4+3 |
| 1-1 | 3+1 | 5-1 | 0+3 | 4+3 |

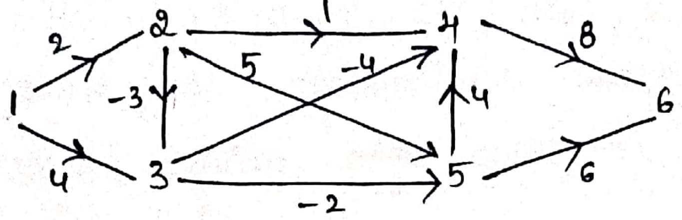
Shortest path tree is:-



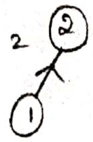

| Vertex | shortest path | Shortest distance |
|---|---|---|
| 2 | 1-4-3-2 | 1 |
| 3 | 1-4-3 | 3 |
| 4 | 1-4 | 5 |
| 5 | 1-4-3-2-5 | 0 |
| 6 | 1-4-6 | 4 |
| 7 | 1-4-3-2-5-7 | 3 |

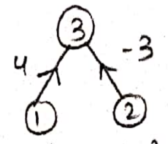
Scanned with OKEN Scanner

Example 2:

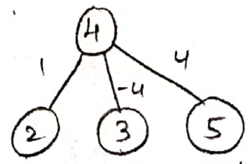


| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 → | 0 | 2 | 4 | ∞ | ∞ | ∞ |
| 2 → | 0 | 2 | -1 | 0 | 2 | ∞ |
| 3 → | 0 | 2 | -1 | -5 | -3 | 8 |
| 4 → | 0 | 2 | -1 | -5 | -3 | 3 |
| 5 → | 0 | 2 | -1 | -5 | -3 | 3 |
| 6 → | 0 | 2 | -1 | -5 | -3 | 3 |

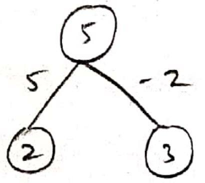




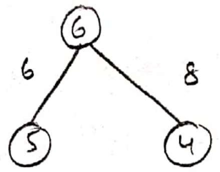
| | |
|---|---|
| 0+4 | 2-3 |
| 0+4 | 2-3 |
| 0+4 | 2-3 |
| 0+4 | 2-3 |
| 0+4 | 2-3 |
| 0+4 | 2-3 |



| | | |
|---|---|---|
| 2+1 | 4-4 | ∞+4 |
| 2+1 | -1-4 | 2+4 |
| 2+1 | -1-4 | -3+4 |
| 2+1 | -1-4 | -3+4 |
| 2+1 | -1-4 | -3+4 |
| 2+1 | -1-4 | -3+4 |



| | |
|---|---|
| 2+5 | 4-2 |
| 2+5 | -1-2 |
| 2+5 | -1-2 |
| 2+5 | -1-2 |
| 2+5 | -1-2 |
| 2+5 | -1-2 |



| | |
|---|---|
| ∞+6 | ∞+8 |
| 0+8 | 2+6 |
| -5+8 | -3+6 |
| -5+8 | -3+6 |
| -5+8 | -3+6 |
| -5+8 | -3+6 |

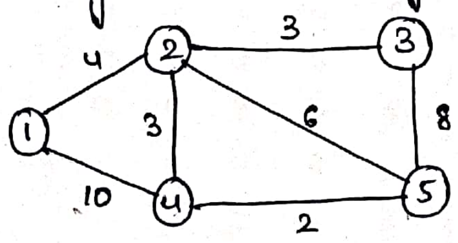
Scanned with OKEN Scanner

Algorithm BellmanFord (v, cost, dist, n)
// single source / all destinations shortest
// Paths with negative edge costs.
{
for i := 1 to n do
    dist[i] := cost[v, i];
for k := 2 to n-1 do
    for each u such that u ≠ v and u has atleast one incoming edge do
        for each (i, u) in the graph, do
            if dist[u] > dist[i] + cost[i, u], then
                dist[u] := dist[i] + cost[i, u];
}

T.C = $O(ve)$ or $O(n^3)$

* Floyd - Warshall Algorithm
[All path shortest path problem]

Consider the following graph.




The weight matrix of the given graph is :-

$$W = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & \infty & 10 & \infty \\ 4 & 0 & 3 & 3 & 6 \\ \infty & 3 & 0 & \infty & 8 \\ 10 & 3 & \infty & 0 & 2 \\ \infty & 6 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

$$D^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & \infty & 10 & \infty \\ 4 & 0 & 3 & 3 & 6 \\ \infty & 3 & 0 & \infty & 8 \\ 10 & 3 & \infty & 0 & 2 \\ \infty & 6 & 8 & 2 & 0 \end{bmatrix} \end{array} ; \quad D^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 4 & \infty & 10 & \infty \\ 4 & 0 & 3 & 3 & 6 \\ \infty & 3 & 0 & \infty & 8 \\ 10 & 3 & \infty & 0 & 2 \\ \infty & 6 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

Scanned with OKEN Scanner

$$D^2 = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 4 & 7 & 7 & 10 \\ 4 & 0 & 3 & 3 & 6 \\ 7 & 3 & 0 & 6 & 8 \\ 7 & 3 & 6 & 0 & 2 \\ 10 & 6 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

31: 32 21

34: 32 24, $3 + 3 = 6$

35: 32 25, $3 + 6 = 9$

41: 42 21, $3 + 4 = 7$

45: 42 25, $3 + 6 = 9$

$$D^3 = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 4 & 7 & 7 & 10 \\ 4 & 0 & 3 & 3 & 6 \\ 7 & 3 & 0 & 6 & 8 \\ 7 & 3 & 6 & 0 & 2 \\ 10 & 6 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

12: 13 32, $7 + 3 = 10$

14: 13 34, $7 + 6 = 13$

15: 13 35, $7 + 8 = 15$

24: 23 34, $3 + 6 = 9$

25: 23 35, $3 + 8 = 11$

45: 43 35, $6 + 8$

$$D^4 = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 4 & 7 & 7 & 9 \\ 4 & 0 & 3 & 3 & 5 \\ 7 & 3 & 0 & 6 & 8 \\ 7 & 3 & 6 & 0 & 2 \\ 9 & 5 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

$$D^5 = \begin{array}{c|c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 4 & 7 & 7 & 9 \\ 4 & 0 & 3 & 3 & 5 \\ 7 & 3 & 0 & 6 & 8 \\ 7 & 3 & 6 & 0 & 2 \\ 9 & 5 & 8 & 2 & 0 \end{bmatrix} \end{array}$$

Scanned with OKEN Scanner

15-09-2025 Monday

* Floyd - Warshall Algorithm:-

// w: weight matrix of graph

// n: number of vertices

Algorithm Floyd-Warshall (w, n)

{

    D := w;

    for k=1 to n

        for i=1 to n

            for j=1 to n

                ~~D[i][j]~~ D[i,j] = min{D[i,j], D[i,k] + D[k,j]};

    return (D);

}

17-09-2025 Wednesday

* Explain 0/1 Knapsack problem using dynamic programming

* Explain merging and perging rules in 0/1 of knap sack problem.

Capacity $C = 30$

| i | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P_i$ | 2 | 5 | 8 | 1 |
| $W_i$ | 10 | 15 | 6 | 9 |

$S = \{(P, w) \,/\, P\text{-profit}, w\text{-weight}\}$

$S^0, S^1, S^2, S^3, S^4$

1) $S^0 = \{(0,0)\}$

Item 1: $P = 2$ $\quad w = 10$

$S_1^0 \quad \{(0+2, 0+10)\} = \{(2,10)\}$

$\boxed{S^1 = S^0 \cup S_1^0}$

$= \{(0,0)(2,10)\}$

Item 2 : $P = 5$ , $w = 15$

$$S_1^1 = \{(0+5, 0+15), (2+5, 10+15)\}$$
$$= \{(5, 15), (7, 25)\}$$

$$\boxed{S^2 = S^1 \cup S_1^1}$$

$$= \{(0,0)\ (2,10)\ (5,15)\ (7,25)\}$$

Item 3 : $P = 8$ , $w = 6$

$$S_1^2 = \{(0+8, 0+6)(2+8, 10+6)(5+8, 15+6)(7+8, 25+6)\}$$
$$= \{(8,6)\ (10,16)\ (13,21)\ (15,31)\}$$

$$\boxed{S^3 = S^2 \cup S_1^2}$$

$$S^3 = \{(0,0)(2,10)(5,15)(7,25)(8,6)(10,16)(13,21)(15,31)\}$$

→ For the pair $(15, 31)$, $w = 31 >$ knapsack capacity so discard it.

→ The pairs $(2,10)$, $(5,15)$, $(7,25)$ dominated by $(8,6)$.

So discard them This is called purging

→ Hence $S^3 = \{(0,0)(8,6)(10,16)(13,21)\}$

Item 4: $p = 1$ , $w = 9$

$$S_1^3 = \{(0+1, 0+9)(8+1, 6+9)(10+1, 16+9)(13+1, 21+9)\}$$
$$= \{(1,9)(9,15)(11,25)(14,30)\}$$

$$\boxed{S^4 = S^3 \cup S_1^3} \rightarrow \text{called merging}$$

$$= \{(0,0)(8,6)(10,16)(13,21)(1,9)(9,15)(11,25),(14,30)\}$$

Optimal profit $= 14$ and $(x_1\ x_2\ x_3\ x_4) = (0, 1, 1, 1)$

Scanned with OKEN Scanner

*

| $i$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P_i$ | 12 | 10 | 20 | 15 |
| $W_i$ | 2 | 1 | 3 | 2 |

$S^0 = \{(0,0)\}$

**Item 1:** $p = 12$, $\omega = 2$

$S_1^0 = \{(0+12, 0+2)\} = \{(12,2)\}$

$\boxed{S^1 = S^0 \cup S_1^0}$

$S^1 = \{(0,0)(12,2)\}$

**Item 2:** $p = 10$, $\omega = 1$

$S_1^1 = \{(0+10, 0+1), (12+10, 2+1)\}$

$= \{(10,1)(22,3)\}$

$\boxed{S^2 = S^1 \cup S_1^1}$

$= \{(0,0)(12,2)(10,1)(22,3)\}$

**Item 3:** $p = 20$, $\omega = 3$

$S_1^2 = \{(0+20, 0+3)(12+20, 3+2)(10+20, 1+3)(22+20, 3+3)\}$

$\begin{array}{r} 20 \\ 12 \\ \hline 32 \end{array}$ $= \{(20,3)(32,5)(30,4)(42,6)\}$

$\begin{array}{r} 22 \\ 20 \\ \hline 42 \end{array}$ $\boxed{S^3 = S^2 \cup S_1^2}$

$S^3 = \{(0,0)(12,2)(10,1)(22,3)\cancel{(20,3)}(32,5)(30,4)\cancel{(42,6)}\}$

→ For the pair $(42,6)$, $\omega = 6 >$ knapsack capacity so discard it.

→ The pairs $(20,3)$ dominated by $(22,3)$ so discard them.

→ Hence $S^3 = \{(0,0)(12,2)(10,1)(22,3)(32,5)(30,4)\}$

Scanned with OKEN Scanner

Item 4: $P = 15$ , $w = 2$

$$S_1^3 = \{(0+15, 0+2)(12+15, 2+2)(10+15, 1+2)(22+15, 3+2)(32+15, 5+2)(30+15, 4+2)\}$$

$$= \{(15,2)(27,4)(25,3)(37,5)\cancel{(47,7)}\cancel{(45,6)}\}$$

$$\boxed{S^4 = S^3 \cup S_1^3}$$

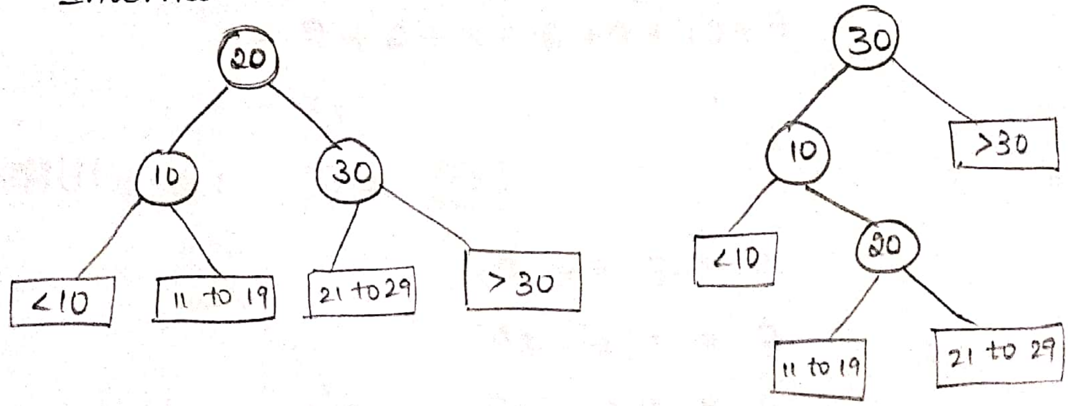
$$S^4 = \{(0,0)(10,1)(30,4)(15,2)(25,3)(37,5)\}$$

optimal profit $= 37$ and $(x_1\ x_2\ x_3\ x_4) = (1,1,0,1)$

*what is optimal Binary Search tree explain?

Process binary search tree for 3 internal nodes:

Internal nodes : 10, 20, 30




Internal nodes : $a_1, a_2, a_3$ | $a_1\ a_2\ a_3 \cdots\cdots a_n$

External nodes : $E_0\ E_1\ E_2\ E_3$ | $E_0\ E_1\ E_2\ E_3 \cdots\cdots E_n$

$P(i)$: The probability (frequency) with which we search for $a_i$

$q(i)$: The probability (frequency) with which we search for $E_i$

cost for internal node: $P(i) * \text{level}(a_i)$

cost for external node: $q(i) * [\text{level}(E_i) - 1]$

cost for a BST is $\sum_{i=1}^{n} P(i) * \text{level}(a_i) + \sum_{i=0}^{n} q(i) * [\text{level}[E_i] - 1]$

Conclusion:

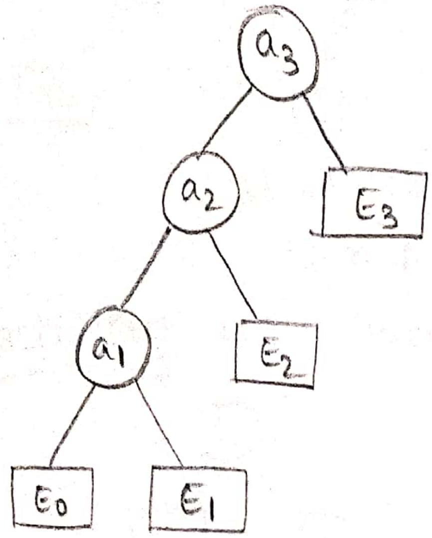
$\therefore$ The Binary search tree with minimum possible cost is called optimal Binary Search tree.

Scanned with OKEN Scanner

* Example:- $(P_1, P_2, P_3) = (3, 1, 8)$

$(q_0, q_1, q_2, q_3) = (2, 3, 5, 7)$

Possibility 1 : Tree $\boxed{T1}$




cost (T1) = $a_1 \rightarrow 3 \times 3$

$a_2 \rightarrow 1 \times 2$

$a_3 \rightarrow 8 \times 1$

$E_0 \rightarrow 2 \times (4-1)$

$E_1 \rightarrow 3 \times (4-1)$

$E_2 \rightarrow 5 \times (3-1)$

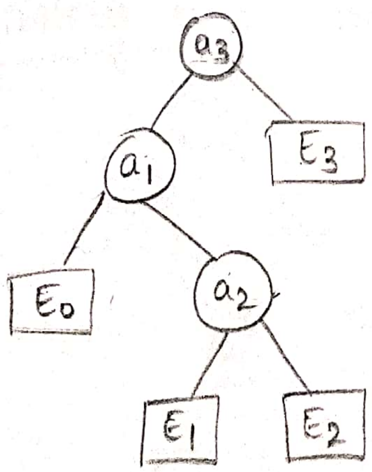
$E_3 \rightarrow 7 \times (2-1)$

$\text{cost (T1)} = 3\times3 + 1\times2 + 8\times1 + 2\times3 + 3\times3 + 5\times2 + 7\times1$

$= 9 + 2 + 8 + 6 + 9 + 10 + 7$

$= 51$

Possibility 2 : Tree $\boxed{T2}$




$a_1 \rightarrow 3 \times 2$

$a_2 \rightarrow 1 \times 3$

$a_3 \rightarrow 8 \times 1$

$E_0 \rightarrow 2 \times (3-1)$

$E_1 \rightarrow 3 \times (4-1)$

$E_2 \rightarrow 5 \times (4-1)$

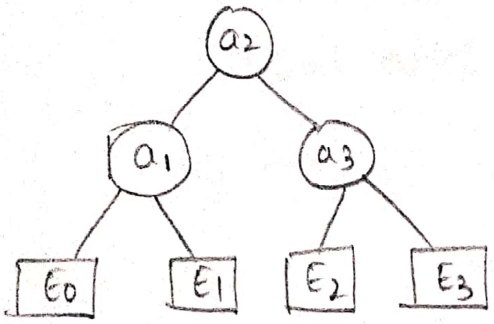
$E_3 \rightarrow 7 \times (2-1)$

$\text{cost} = 3\times2 + 1\times3 + 8\times1 + 2\times2 + 3\times3 + 5\times3 + 7\times1$

$= 52$

Scanned with OKEN Scanner

Possibility 3 : Tree $\boxed{T_3}$



$a_1 \rightarrow 3 \times 2$ $\qquad E_0 \rightarrow 2 \times (3-1)$

$a_2 \rightarrow 1 \times 1$ $\qquad E_1 \rightarrow 3 \times (3-1)$

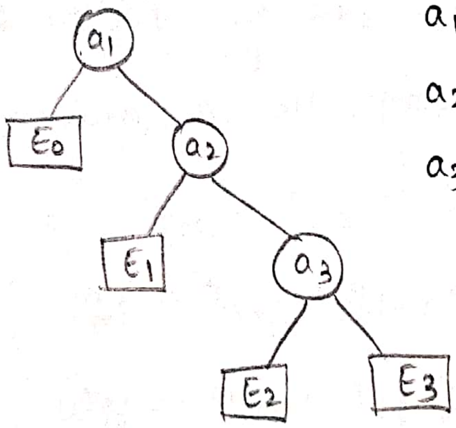
$a_3 \rightarrow 8 \times 2$ $\qquad E_2 \rightarrow 5 \times (3-1)$

$\qquad E_3 \rightarrow 7 \times (3-1)$

$$\text{cost} = 3\times2 + 1\times1 + 8\times2 + 2\times2 + 3\times2 + 5\times2 + 7\times2$$

$$= 57$$

Possibility 4 : Tree $\boxed{T_4}$



$a_1 \rightarrow 3 \times 1$ $\qquad E_0 \rightarrow 2 \times (2-1)$

$a_2 \rightarrow 1 \times 2$ $\qquad E_1 \rightarrow 3 \times (3-1)$

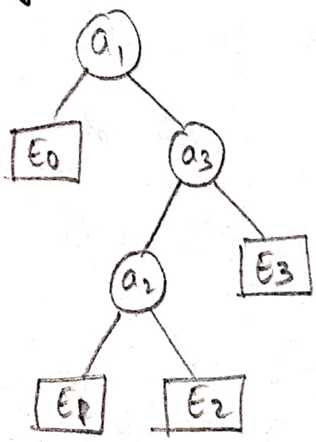
$a_3 \rightarrow 8 \times 3$ $\qquad E_2 \rightarrow 5 \times (4-1)$

$\qquad E_3 \rightarrow 7 \times (4-1)$

$$\text{cost}(T_4) = 3\times1 + 1\times2 + 8\times3 + 2\times1 + 3\times2 + 5\times3 + 7\times3$$

$$= 73$$

Possibility 5 : Tree $\boxed{T_5}$



$a_1 \rightarrow 3 \times 1$ $\qquad E_0 \rightarrow 2 \times (2-1)$

$a_2 \rightarrow 1 \times 3$ $\qquad E_1 \rightarrow 3 \times (4-1)$

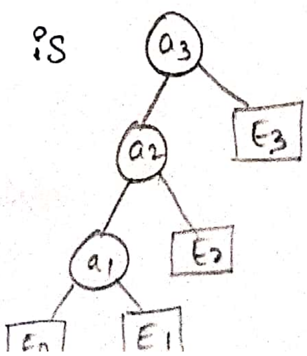
$a_3 \rightarrow 8 \times 2$ $\qquad E_2 \rightarrow 5 \times (4-1)$

$\qquad E_3 \rightarrow 7 \times (3-1)$

$$\text{cost}(T_5) = 3\times1 + 1\times3 + 8\times2 + 2\times1 + 3\times3 + 5\times3 + 7\times2$$

$$= 62$$

$\therefore$ Optimal Binary Search tree is



and optimal cost is 51.

8-04-... Thursday

* Optimal Binary Search Tree

Internal nodes : $a_1, a_2, a_3, \dots, a_n$

External nodes : $E_0, E_1, E_2, E_3, \dots, E_n$

$t_{ij} \rightarrow$ BST which contains

$a_{i+1} \quad a_{i+2} \dots\dots a_j$

$E_i \quad E_{i+1} \quad E_{i+2} \dots\dots E_j$

$t_{0n} \rightarrow \boxed{\begin{matrix} & a_1 & a_2 \dots\dots a_n \\ E_0 & E_1 & E_2 \dots\dots E_n \end{matrix}}$

$\rightarrow w_{ij}$ : weight of the tree $t_{ij}$

= Sum of all probabilities of nodes in $t_{ij}$

$= P_{i+1} + P_{i+2} + \dots\dots P_{j-1} + P_j$

$q_i + q_{i+1} + q_{i+2} + \dots\dots q_{j-1} + q_j$

$$w_{ij} = w_{ij-1} + [P_j + q_j]$$

$$w_{ii} = q_i$$

$\rightarrow c_{ij}$ – cost of tree $t_{ij}$ :

$a_4 \quad a_5 \dots\dots a_{10}$

$a_{i+1} \quad a_{i+2} \dots\dots a_k \dots\dots a_j$

So,

$$\begin{array}{ccc} & a_k & \\ \boxed{a_{i+1} \; a_{i+2} \dots a_{k-1}} & & \boxed{a_{k+1} \; a_{k+2} \dots a_j} \\ t_{ik-1} & & t_{kj} \\ C_{ik-1} & & \end{array}$$

$$c_{ij} = \min_{i < k \leq j} \{ c_{ik-1} + c_{kj} \} + w_{ij}$$

$c_{ii} = 0$

Scanned with OKEN Scanner

$\therefore$ $r_{ij}$ = root of the tree $t_{ij}$

= The K value for which $c_{ij}$ is minimum

$r_{ij} = 0$.

Q: Find Optimal Binary Search tree for identifiers $(a_1\ a_2\ a_3\ a_4)$ = (for while, if, end) with probabilities / frequencies $(P_1\ P_2\ P_3\ P_4)$ = (1, 4, 2, 1) and $(q_0\ q_1\ q_2\ q_3\ q_4)$ = (4, 2, 4, 1, 1)

Given,

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | |
|---|---|---|---|---|---|
| $P \rightarrow$ | 1 | 4 | 2 | 1 | |
| $q \rightarrow$ 4 | 2 | 4 | 1 | 1 | |
| $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | |

Formulas:

~~$w_{ij} = w_{ij} + [P_{ij}$~~

1) $W_{ij} = W_{ij-1} + [P_j + q_j]$ , $W_{ii} = q_i$

2) $C_{ij} = \min\limits_{K=i+1 \text{ to } j} \{ C_{iK-1} + C_{Kj} \} + W_{ij}$ , $C_{ii} = 0$

3) $r_{ij}$ = root of tree $t_{ij}$ , $\boxed{r_{ii} = 0}$

| | | | | |
|---|---|---|---|---|
| $W_{00} = 4$<br>$C_{00} = 0$<br>$r_{00} = 0$ | $W_{11} = 2$<br>$C_{11} = 0$<br>$r_{11} = 0$ | $W_{22} = 4$<br>$C_{22} = 0$<br>$r_{22} = 0$ | $W_{33} = 1$<br>$C_{33} = 0$<br>$r_{33} = 0$ | $W_{44} = 1$<br>$C_{44} = 0$<br>$r_{44} = 0$ |
| $W_{01} = 7$<br>$C_{01} = 7$<br>$r_{01} = 1$ | $W_{12} = 10$<br>$C_{12} = 10$<br>$r_{12} = 2$ | $W_{23} = 7$<br>$C_{23} = 7$<br>$r_{23} = 3$ | $W_{34} = 3$<br>$C_{34} = 3$<br>$r_{34} = 4$ | |
| $W_{02} = 15$<br>$C_{02} = 22$<br>$r_{02} = 2$ | $W_{13} = 13$<br>$C_{13} = 20$<br>$r_{13} = 2$ | $W_{24} = 9$<br>$C_{24} = 12$<br>$r_{24} = 3$ | | |
| $W_{03} = 18$<br>$C_{03} = 32$<br>$r_{03} = 2$ | $W_{14} = 15$<br>$C_{14} = 27$<br>$r_{14} = 2$ | | | |
| $W_{04} = 20$<br>$C_{04} = 39$<br>$r_{04} = 2$ | | | | |

Scanned with OKEN Scanner

$\rightarrow w_{00} = q_{0} = 4$ ; $w_{11} = q_1 = 2$ ; $w_{22} = q_2 = 4$

$w_{33} = q_3 = 1$ ; $w_{44} = q_4 = 1$

$\rightarrow w_{01} = w_{00} + [P_1 + q_1]$

$= 4 + [1+2] = 7$

$i=0,\ j=1,\ k=1$

$C_{01} = \min\{C_{00} + C_{11}\} + w_{01}$

$= \min\{0+0\} + 7 = 7$

$r_{01} = 1$

$\rightarrow w_{12} = w_{11} + [P_2 + q_2]$

$= 2 + [4+4] = 10$

$C_{12} = 0 + 10$

$= 10$

$\rightarrow w_{13} = w_{12} + [P_3 + q_3]$

$= 10 + [2+1]$

$= 10 + 3 = 13$

$\cancel{C_{13} = \min\{0+7,\ 10+4+13+2\}}$

$= \cancel{20}$

$r_{13} = 2$

$\rightarrow w_{23} = w_{22} + [P_3 + q_3]$

$= 4 + [2+1] = 7$

$C_{23} = 0 + w_{23} = 7$

$r_{23} = 3$

$w_{24} = w_{23} + [P_4 + q_4]$

$= 7 + [1+1]$

$= 7 + 2 = 9$

$\rightarrow w_{34} = w_{33} + [P_4 + q_4]$

$= 1 + [1+1]$

$= 3$

$\rightarrow w_{02} = w_{01} + [P_2 + q_2]$

$= 7 + [4+4] = 15$

$C_{02} = \min\{C_{00} + C_{12},\ C_{01} + C_{22}\} + w_{02}$

$= \min\{0+7,\ 0+10\} + 15$

$= 7 + 15$

$= 22$

$r_{02} = 1$

Scanned with OKEN Scanner

$\rightarrow w_{03} = w_{02} + [p_1 + q_1]$

$= 15 + [1 + 2]$

$= 15 + 3$

$= 18$

$C_{03} = \min\{0+20, 7+7, 22+0\} + 18$

$= 14 + 18$

$= 32$

$\rightarrow w_{14} = w_{13} + [p_4 + q_4]$

$= 13 + [1 + 1]$

$= 15$

$C_{14} = \min\{0+12, 10+3, 20+0\} + 15$

$= 12 + 15$

$= 27$

Scanned with OKEN Scanner

-09-2025 Monday

# * LCS [Largest common subsequence]:

Sequence 1 : a b b a b a b

Sequence 2 : a a b b a a a b

Subsequences : aaa , ababab , abbabb

Subsequences : aaa , aaaa , abbab , abab

common subsequences : aaa , abb , bbb , abab , abba , aaab , abbaa , aabab , abbab , bbaab

$\boxed{abbaab}$ length of LCS = 6 .

* Find the length of largest common subsequence:-

Sequence 1 : x y x x y x x

Sequence 2 : x x y y x y y x

Subsequences: xxx , xyxyxy , xxxy

Subsequences: xxx , xxxx , xyxyxyxy .

common sequences:-

xxxx , xyxy , xyyxx , xyxyx , xxxyx , xxyxx .

∴ length of LCS = 5

* $X : x_1 \; x_2 \; x_3 \; \dots\dots \; x_m$

$Y : y_1 \; y_2 \; y_3 \; \dots\dots \; y_n$

$L(i, j)$ : Length of LCS to $x_1 x_2 \dots\dots x_i$ , $y_1 y_2 \dots\dots y_j$

$L(3,5)$ : $\boxed{x_1 x_2 x_3}$ $\boxed{y_1 y_2 y_3 y_4 y_5}$

we need to find $L(m, n)$

$L(0, j) = 0$ for all $j$

$L(i, 0) = 0$ for all $i$

$m = 4$

| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|
| | | | b |

$n = 6$

| | | | |
|---|---|---|---|
| $m_1$ | $m_2$ | $m_3$ | $m_4$ |

| | | | | | |
|---|---|---|---|---|---|
| $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ |

$L(0, j) = 0$ for all $j$

Scanned with OKEN Scanner

$L(i, j) = 1 + L(i-1, j-1) \quad \text{if} \quad x_i = y_j$

$= \max\{L(i-1, j),\ L(i, j-1)\} \quad \text{if} \quad x_i \neq y_j$

$L(0, j) = 0$ for all $j$

$L(i, 0) = 0$ for all $i$

Q: Find the length of LCS for the following using DP.

seq1 :- abbabab; seq2 :- aabbaaab

$L[i, j] = 1 + L[i-1, j-1] \quad \text{if} \quad x[i] = y[j]$

$= \max\{L[i-1, j],\ L[i, j-1]\} \quad \text{if} \quad x[i] \neq y[j]$

$L(0, j) = 0$ for all $j$

$L(i, 0) = 0$ for all $i$

| | 0 | 1 a | 2 a | 3 b | 4 b | 5 a | 6 a | 7 a | 8 b |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| a 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| b 2 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| b 3 | 0 | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 3 |
| a 4 | 0 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| b 5 | 0 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 5 |
| a 6 | 0 | 1 | 2 | 2 | 3 | 4 | 5 | 5 | 5 |
| b 7 | 0 | 1 | 2 | 3 | 3 | 4 | 5 | 5 | 6 |

Q: Find the length of LCS using dynamic programming

seq 1 :- x x y x x y      Seq 2 :- y x y x y

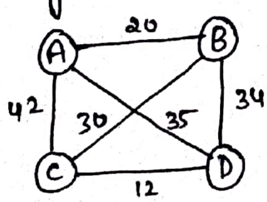
| | | 0 | y 1 | x 2 | y 3 | x 4 | y 5 |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| x | 2 | 0 | 0 | 1 | 1 | 2 | 2 |
| y | 3 | 0 | 1 | 1 | 2 | 2 | 3 |
| x | 4 | 0 | 1 | 2 | 2 | 3 | 3 |
| x | 5 | 0 | 1 | 2 | 2 | 3 | 3 |
| y | 6 | 0 | 1 | 2 | 3 | 3 | 4 |

* Travelling Salesperson problem:-

Q: what is travelling sales person problem. Explain with example

A tour of G is a directed simple cycle that includes
ery vertex in V. The cost of a tour is the sum of the cost
f the edges on the tour. The travelling saleperson problem
is to find the minimum cost.

Eg:-




optimal tour is:-

$A \xrightarrow{20} B \xrightarrow{30} C \xrightarrow{12} D \xrightarrow{35} A$

and cost is 97.

we need to find $g(1, V-\{1\})$

$1 \longrightarrow 5$ — $\boxed{V-\{1,5\}}$

$g(1, V-\{1\})$

$\min\{C_{1K} + g(K, V-\{1,K\})\}$   $2 \le K \le n$

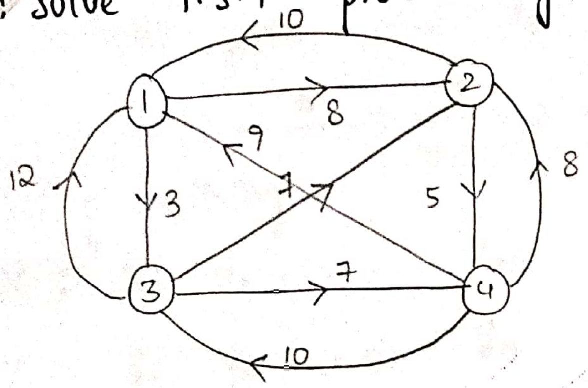
$g(i,S) = \quad i \longrightarrow j \longrightarrow \boxed{S-\{i\}}$

$$g(i,S) = \min \{ c_{ij} + g(j, S-\{j\}) \} \quad j \in S$$

$$g(i,\phi) = c_{i1}$$

$g(i,S) \longrightarrow$ start with vertex $i$, cover all vertices of $S$, end with vertex 1.

Q: Solve T.S.P problem for following problem using DP.




cost matrix

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & 6 & \infty \\ 10 & 0 & \infty & 5 \\ 12 & 7 & 0 & 7 \\ 9 & 8 & 10 & 0 \end{bmatrix} \end{array}$$

$g(2,\phi) = C_{21} = 10$

$g(3,\phi) = C_{31} = 12$

$g(4,\phi) = C_{41} = 9.$

$g(2,\{3\}) = C_{23} + g(3,\phi)$
$= \infty + 12 = \infty$

$g(2,\{4\}) = C_{24} + g(4,\phi)$
$= 5 + 9 = 14$

$g(3,\{2\}) = C_{32} + g(2,\phi)$
$= 7 + 10 = 17$

$g(3,\{4\}) = C_{34} + g(4,\phi)$
$= 7 + 9 = 16$

$g(4,\{2\}) = C_{42} + g(2,\phi)$
$= 8 + 10 = 18$

$g(4,\{3\}) = C_{43} + g(3,\phi)$
$= 10 + 12 = 22.$

$g(2,\{3,4\}) = \min\{ C_{23} + g(3,\{4\}) ; C_{24} + g(4,\{3\}) \}$
$= \min\{ \infty + 16 , 5 + 22 \} = 27.$

$g(3,\{2,4\}) = \min\{ C_{32} + g(2,\{4\}) ; C_{34} + g(4,\{2\}) \}$
$= \min\{ 7 + 14 , 7 + 18 \} = 21.$

Scanned with OKEN Scanner

$g(4, \{2,3\}) = \min \{ C_{42} + g(2, \{3\}) ; C_{43} + g(3, \{2\}) \}$

$= \min \{ 8+\infty , 10+17 \} = 27.$

$g(1, \{2,3,4\}) = \min \{ C_{12} + g(2, \{3,4\}) ; C_{13} + g(3, \{2,4\}) ;$

$C_{14} + g(4, \{2,3\}) \}$

$= \min \{ 8+27 , 6+21 , \infty + 27 \}$

$= 27.$

Optimal path cost is 27 and optimal path is:-

$1 \xrightarrow{6} 3 \xrightarrow{7} 2 \xrightarrow{5} 4 \xrightarrow{9} 1$

* String editing:-

$X : x_1 \; x_2 \; \ldots\ldots \; x_m$

$Y : y_1 \; y_2 \; \ldots\ldots \; y_n$

$\text{cost}(i,j) : x_1 \, x_2 \ldots x_i$
$\quad y_1 \, y_2 \ldots y_j$

$I(y_j)$ — cost of insert operation of $y_j$ into X.

$D(x_i)$ — cost of delete operation of $x_i$ from X.

$C(x_i, y_j)$ — cost of change operation of $x_i$ into $y_j$ in X.

we need to find cost (m,n)

$\text{Cost}(0,0) = 0$

$\text{Cost}(0,j) = \text{cost}(0, j-1) + I(y_j)$

$\text{Cost}(i,0) = \text{cost}(i-1, 0) + D(x_i)$

$\text{cost}(i,j) = \min \begin{cases} \text{cost}(i-1,j) + D(x_i) \\ \text{cost}(i,j-1) + I(y_j) \\ \text{cost}(i-1,j-1) + C(x_i - y_j) \end{cases}$

Q: Find optimal cost of converting X to Y.

X : a b a b a

Y : a a b a

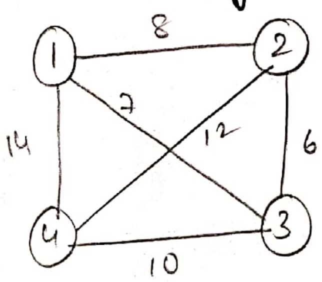
Assume cost of insert and delete operation is 1 and change operation is 2.

| $X\downarrow$ \ $Y\rightarrow$ | | | a | a | b | a |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| | 0 | 0 | 1 | 2 | 3 | 4 |
| a | 1 | 1 | 0 | 1 | 2 | 3 |
| b | 2 | 2 | 1 | 2 | 1 | 2 |
| a | 3 | 3 | 2 | 1 | 2 | 1 |
| b | 4 | 4 | 3 | 2 | 1 | 2 |
| a | 5 | 5 | 4 | 3 | 2 | 1 |

$\searrow +C \quad \downarrow +0 \quad \xrightarrow[+I]{}$

Optimal cost of converting X to Y is 1.

Q: Solve T.S.P for following graph using DP.




$$C = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & 7 & 14 \\ 2 & 8 & 0 & 6 & 12 \\ 3 & 7 & 6 & 0 & 10 \\ 4 & 14 & 12 & 10 & 0 \end{array}$$

$g(2,\phi) = C_{21} = 8$

$g(3,\phi) = C_{31} = 7$

$g(4,\phi) = C_{41} = 14.$

$g(2,\{3\}) = C_{23} + g(3,\phi) = 6 + 7 = 13$

$g(2,\{4\}) = C_{24} + g(4,\phi) = 12 + 14 = 26$

$g(3,\{2\}) = C_{32} + g(2,\phi) = 6 + 8 = 14$

$g(3,\{4\}) = C_{34} + g(4,\phi) = 10 + 14 = 24$

$g(4,\{2\}) = C_{42} + g(2,\phi) = 12 + 8 = 20$

$g(4,\{3\}) = C_{43} + g(3,\phi) = 10 + 7 = 17.$

$g(2,\{3,4\}) = \min\{C_{23} + g(3,\{4\})\ ;\ C_{24} + g(4,\{3\})\}$

$= \min\{6+24,\ 12+17\} = 29.$

$$g(3,\{2,4\}) = \min\{C_{32} + g(2,\{4\}) ; C_{34} + g(4,\{2\})\}$$

$$= \min\{6+26, 10+20\}$$

$$= 30$$

$$g(4,\{2,3\}) = \min\{C_{42} + g(2,\{3\}) ; C_{34} + g(3,\{2\})\}$$

$$= \min\{12+13, 10+14\} = 24$$

$$g(1,\{2,3,4\}) = \min\{C_{12} + g(2,\{3,4\}) ; C_{13} + g(3,\{2,4\}) ; C_{14} + g(4,\{2,3\})\}$$

$$= \min\{8+29, 7+30, 14+24\}$$

$$= 37.$$

Optimal cost is 37:- 1 — 2 — 4 — 3 — 1

Scanned with OKEN Scanner