25/9/21

DYNAMIC PROGRAMMING

Fibonacci number -

$F(n) = F(n-1) + F(n-2)$ for $n > 1$

→ Dynamic programming is a technique for solving problems with overlapping subproblems.

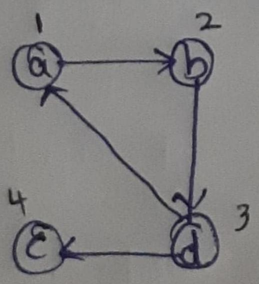
→ These subproblems arise from a recurrence relating a given problem's solution to solutions of its smaller problems.

→ PRINCIPLE OF OPTIMALITY -

An optimal solution to any instance of an optimization problem is composed of optimal solutions to its sub-instances.

⇒ Warshall's Algorithm -

The transitive closure of a directed graph with n vertices can be defined as the $n \times n$ boolean matrix $T = \{t_{ij}\}$, in which the element in the i^{th} row and the j^{th} column is 1 if there exists a non trivial path (i.e, directed path of a positive length) from the i^{th} vertex to the j^{th} vertex, otherwise t_{ij} is 0.



$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$R^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

R_0 - Path matrix with no intermediate vertices.

$$R^{(1)} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

R_1 - Path matrix with intermediate vertices numbered not > 1

~~R2~~ - R^{2} - path matrix with intermediate vertices numbered not > 2.

$$R^{2} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$R^{3} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

R_3 - path matrix with intermediate vertices numbered not > 3.

$$R^{4} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

R_4 - path matrix with intermediate vertices numbered not > 4.

$\Rightarrow$ v_i, a list of intermediate vertices each numbered not higher than k, v_j.

v_i, vertices numbered $\leq k-1$, v_k, vertices numbered $\leq k-1$, v_j.

$\boxed{v_i < k < v_k < k - v_j}$

$$r_{ik}^{(k-1)} = 1 \qquad r_{kj}^{(k-1)} = 1$$

$r_{ij}^{K} = r_{ij}^{(K-1)}$ (or) $r_{ik}^{(K-1)}$ and $r_{kj}^{(K-1)}$

→ Warshall's algorithm constructs the transitive closure through a series of n×n boolean matrices:

$$R^{(0)}, \ldots, R^{(K-1)}, R^{(K)} \ldots R^{(n)}.$$

→ Let $r_{ij}^{(K)}$, the element in the ith row and jth column of matrix $R(K)$, be equal to 1.

Algorithm: Warshall (A [1..n, 1..n])

Implements Warshall's algorithm for computing the transitive closure.

Input - The adjacency matrix A of a digraph with n vertices.

Output - The transitive closure of the digraph.

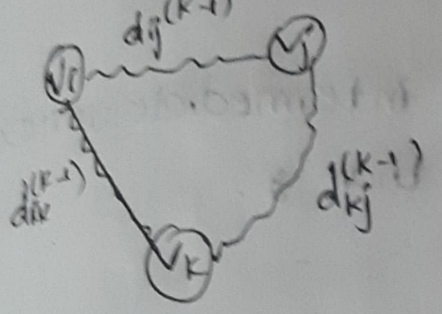
```
R(0) ← A
for k ← 1 to n do
    for i ← 1 to n do
        for j ← 1 to n do.
            R^K[i,j] ← R^(K-1)[i,j] (or)
            (R^(K-1)[i,K] and R^(K-1)[K,j])
return R(n)
```

→ Time Complexity - $O(n^3)$

FLOYD'S ALGORITHM

Floyd's algorithm computes the distance matrix of weighted graph with n vertices through a series of $n \times n$ matrices.

$$D^{(0)}, \ldots D^{(k-1)}, D^{(k)}, \ldots D^{(n)}.$$



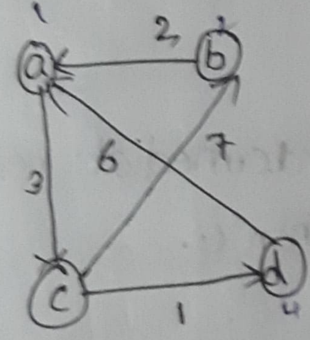
$v_i \rightarrow v_j$

① $d_{ij}^{(k-1)}$

② $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$

$$\min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

eg:



D^0 = Distance matrix with zero intermediate vertices.

$$D^0 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

∞ bcz it doesnot has a direct path so, we donno whether a path do exist or not without '0' intermediate vertices.

D^1 - Distance matrix with 1 intermediate vertex numbered not > 1

$$D^1 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

D^2 - Distance matrix with intermediate vertex numbered not greater than 2.

$$D^2 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ 9 & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

D^3 - Distance matrix with intermediate vertex numbered not greater than 3.

$$D^3 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

D^4 - Distance matrix with intermediate vertex numbered not greater than 4.

$$D^4 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix} \end{array}$$

ALGORITHM Floyd (W[1..n, 1..n])

Implements Floyd's algorithm for all the pairs shortest - paths problem.

Input - The weight matrix W of a graph with no negative - length cycle.

Output - The dist matrix of the shortest path's lengths.

$D \leftarrow W$

for $k \leftarrow 1$ to n do

for $j \leftarrow 1$ to n do

for $i \leftarrow 1$ to n do

$D[i,j] \leftarrow \min\{D[i,j], D[i,k] + D[k,j]\}$

return D.