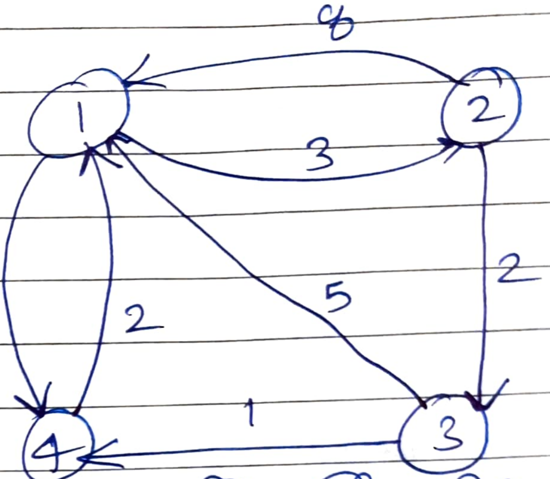
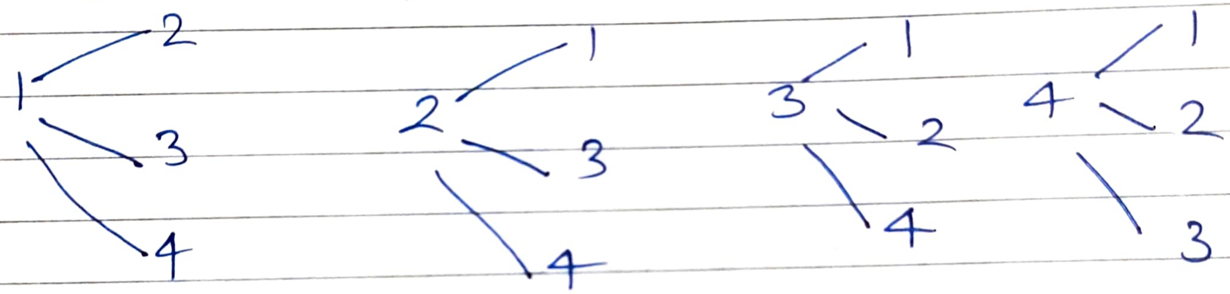
All pairs Shortest Path (Floyd-warshall Algo.

- problem is to find the shortest path between every pair of vertices.



Let us say the starting vertex is ①, ②, ③, ④ respectively.



- So this looks similar to Dijkstra's Algo. But Dijkstra's Algo will find the shortest path from one of the source vertex & it takes $O(n^2)$.

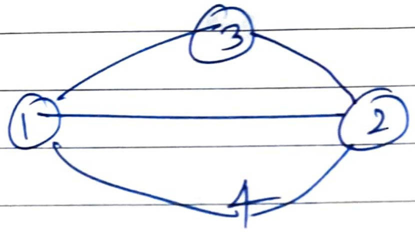
If we run Dijkstra's algo on all the vertices one by one i.e all n vertices, then we can get the results. So time will be $O(n^3)$.

Solving this problem using Dynamic programming.

- Dynamic programming says that the problem should be solved by taking sequence of decision. In each stage we will take a decision. What decision we have to take?

Suppose we have to find shortest path between ① ——— ②

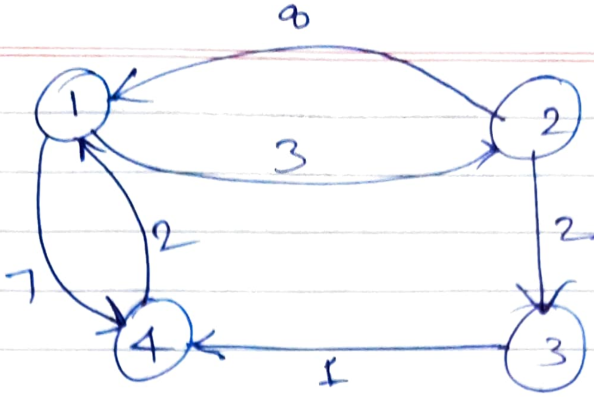
· there can be direct edge path between ① ——— ②, or may be shorter path going via vertex ③, or may be going via vertex ④.



So, in this way we have to check or decide whether the shorter path is going via some other vertex. So we will start selecting the middle vertex as a vertex ①, find out first whether there is a shorter path between all the pair of vertices going via vertex ①, then via vertex ② and so on. So this is how we can take decision, or sequence of decision.

So How this can be done?

this can be done by just preparing matrices.



In case of ~~any~~ absence of any edge then keep ∞ Otherwise for self loop use 0.

$$A^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & \infty & 7 \\ 2 & 8 & 0 & 2 & \infty \\ 3 & 5 & \infty & 0 & 1 \\ 4 & 2 & \infty & \infty & 0 \end{array}$$

original matrix

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 8 & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 8 & \infty & 0 \end{array}$$

Consider Intermediate vertex ①

So is there any better shorter path going via ①

All the path that belong to vertex ① will be remain unchanged so I should not calculate them. Directly we can take.

$$A^0[2,3] \quad A^0[2,1] + A^0[1,3]$$
$$2 < 8 + \infty$$

$$A^0[2,4] \quad A^0[2,1] + A^0[1,4]$$
$$\infty > 8 \quad 7$$

$$A^0[3,2] \quad A^0[3,1] + A^0[1,2]$$
$$\infty > 5 + 3$$

- prepare a matrix going via vertex ②

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 8 & 7 & 0 \end{array}$$

$A^1[1,3] \quad A^1[1,2] + A^1[2,3]$

$\infty > 3 + 2$

$A^1[1,4] \quad A^1[1,2] + A^1[2,4]$

$7 < 3 + 15$

⋮

- prepare a matrix going via vertex ③

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 7 & 0 & 2 & 3 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

$A^2[1,2] \quad A^2[1,3] + A^2[3,2]$

$3 < 5 \quad 8$

⋮

- prepare a matrix going via vertex ④

$$A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 5 & 0 & 2 & 3 \\ 3 & 3 & 6 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

So here we prepared all the matrices. Finally the ④ vertex, when we consider, got the shortest path ~~of~~ between all pair of vertices. So we have taken sequence of decisions in each stage, we were getting the matrix.

So prepare the formula :⇒

- By selecting any vertex as intermediate vertex, then How we were getting the values in the matrix.

$$A^K[i,j] = \min\left\{ \underline{A^{K-1}[i,j]},\ \boxed{A^{K-1}[i,K] + A^{K-1}[K,j]} \right\}$$

K = intermediate vertex
K-1 = is corresponding with previous matrix.

Code:

```
for (k=1, k<=n, k++)
{
 for(i=1, i<=n; i++)
 {
   for (j=1, j<=n, j++)
   {
      A[i,j]= min{A[i,j], A[i,k]+A[k,j]};
   }
 }
}
```

there is three nested for loop, so time will be $O(n^3)$