**Experiment-7**

**To WAP and analyze the All-Pair shortest path problem using Floyd-Warshall**

**algorithm.**

1. **Define the problem.**

Ans. Example:

6

1

9

2

4

3

2

1

5

- 4

Given a weighted directed graph of vertex 4 ( 1 ,2,3,4) and there distances between them as

shown in the figure so we have to find all pair shortest path using (**Floyd-Warshall algorithm.**)

1. **Explain how dynamic approach can be applied to this problem.**

Ans. We have to divide this problem in sequence of sub problem and solved them one by one

We have to create distance matrices for the graph for example D1 ,D2 ,D3 and from

them we have to calculate the optimal solution .

**Floyd-Warshall algorithm:**

**Formula**

Dkth[i][j]=min(D(kth) [i][j], D(k-1)th [i][k]+ D(k-1)th [i][k])

1. **Write pseudo code for this problem and find Time and Space complexity.**

**Ans**. pseudo code**:**

1. Enter the number of vertex in n;
2. Enter the first Distance martix all value
3. For i=0 to n :
4. For j=0 to n;
5. Input D[i][j]
6. For k=0 to n:
7. Print : D" k " matrix:
8. For i=0 to n:
9. For j=0 to n:
10. if ( D[i][k] + D[k][j] < D[i][j]):

true

D[i][j] = D[i][k] + D[k][j];

cout<<D[i][j]<<" ";

else

cout<<D[i][j]<<" ";

11. print : Resultent shortest paths matrix :

12. for i=0 to <n:

13.for j=0 to <n:

Print :D[i][j]<<" ";

14. exit

Time complexity : o(n^3) and space complexity :

1. **Write Source Code in C/C++.**

**Code:**

#include<iostream>

#include <algorithm>

using namespace std;

#define INF 999

main(){

int k=0,i=0,j=0,n,D[50][50],resultent[50][50];

cout<<"\* note take infinty -> 1000\n";

cout<<"Enter the number of vertex(note at max 4 ):";

cin>>n;

cout<<"Enter the D"<<0<<" martix all value:";

for(i=0;i<n;i++){

for(j=0;j<n;j++){

cin>>D[i][j];

}

}

//calculation code

for(k=0;k<n;k++){

cout<<"\nFor D"<<k<<" matrixs:\n";

for(i=0;i<n;i++){

for(j=0;j<n;j++){

if (D[i][k] + D[k][j] < D[i][j]){

D[i][j] = D[i][k] + D[k][j];

cout<<D[i][j]<<" ";

}

else{

cout<<D[i][j]<<" ";

}

}

cout<<endl;

}

}

cout<<endl;

cout<<" Resultent shortest paths matrix :\n";

for(i=0;i<n;i++){

for(j=0;j<n;j++){

cout<<D[i][j]<<" ";

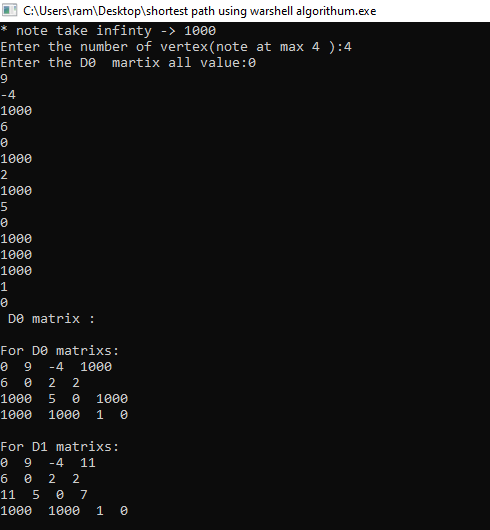
}

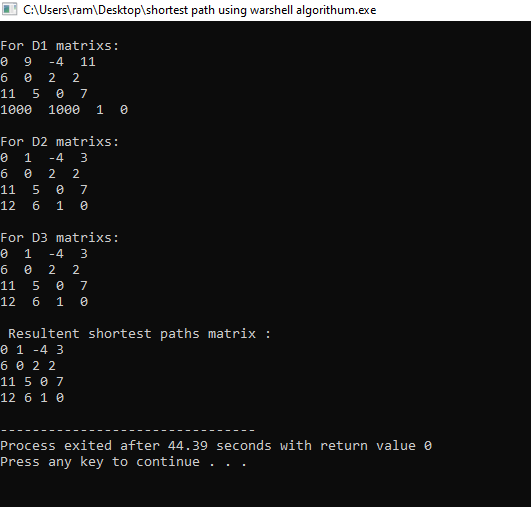
cout<<endl;

}

}

**Output:**

****

****

1. Variables and other data structure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Datatype** | **Typical Value** | **Minimum Value** | **Maximum Value** |
| I | Int | (0 to N) | 0 | N |
| J | Int | (0 to N) | 0 | N |
| k | Int | (0 to N) | 0 | N |
| N | Int | 4 | 0 | - |
| D[50][50] | Int | (0,9,4,INF,6,0,INF,  2,INF,5,0,INF,INF  INF,1,0) | 0 | INF |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Test Plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | **Expected Output** | **Actual Output** | **Comments** |
| (0,9,4,INF,6,0,INF,  2,INF,5,0,INF,INF  INF,1,0) | (0,1,-4,3,6,0,2,2,11,  5,0,7,12,6,1,0) | (0,1,-4,3,6,0,2,2,11,  5,0,7,12,6,1,0) | - |
|  |  |  |  |