Batch: A 1 (Honours) Experiment Number: 2

Roll Number: 16010421059 Name: Chinmay Mhatre

Aim of the Experiment: Implementation of Uninformed search algorithm – DFS

Code:

```
#include <stdio.h>
 int v;
void dfs(int arr[][v], int visited[], int start)
{
   int a[v];
   int flag=0;
   int i=0;
   for(int i=0;i<v;i++) {
      if (start == a[i]) {
         flag++;
      }
   }

   if (flag==0 || start==0)
   {
      printf(" %d ",start);
   }
  for(int i=0;i<v;i++) {
      a[i]=start;
   }
         visited[start] =1;
         for(int i=0;i<v;i++)
         {
                  if(visited[i]!=1 && arr[start][i]==1)
                  {
                           dfs(arr,visited,i);
                  }
         }
}

int main() {
  int i,j,k,adj,start;
  printf("Enter number of vertices: ");
  scanf("%d",&v);
  int arr[v][v];
  int visited[v];
  int open[v];
  for (i = 0; i < v; i++)
  {
   for (j = 0; j < v; j++)
   { arr[i][j] = 0;  }
  }
```

```
 for(i = 0; i < v; i++)
 {
    printf("Enter number of adjacent nodes for vertex %d: ",i);
    scanf("%d",&adj);
    for (k=0;k<adj;k++)
    {
       scanf("%d",&j);
       arr[i][j] = 1;

    }
 }
 printf("\n");
 for (i = 0; i < v; i++) {
  printf("%d: ", i);
  for (j = 0; j < v; j++) {
    printf("%d ", arr[i][j]);
  }
  printf("\n");
 }

 printf("\nEnter the node you want to start from: ");
 scanf("%d",&start);

 for(int i=0;i<v;i++)
  {
  if(arr[start][i]==1)
    {
       open[i]=i;
    }

  }
    printf("\nVisited: %d ",start);
    printf("\nOpen: %d ",open[i]);

    for(i=0;i<v;i++)
    {
    j = open[i];
    if(arr[j][i]==1)
    {
       open[i]=i;
    }
    }

 for(int i =0;i<v;i++)
 {
    visited[i] = 0;
    printf("\nDFS is: ");
dfs(arr,visited, start);}}
```

Output/Result:

```
/tmp/Jwp6IUaG6P.o
Enter number of vertices: 3
Enter number of adjacent nodes for vertex 0: 2
1 2
Enter number of adjacent nodes for vertex 1: 1
0
Enter number of adjacent nodes for vertex 2: 1
0
0: 0 1 1
1: 1 0 0
2: 1 0 0

Enter the node you want to start from: 1
Visited: 1
Open: 0
Visited: 1 0
Open: 2
DFS is:  1  0  2
```

Outcomes:

CO2: Analyse and formulate the problem and select appropriate search method and write the algorithm.

Conclusion (based on the Results and outcomes achieved):

The DFS algorithm was understood and implemented with the help of C programming. An adjacency matrix was derived on the basis of nodes adjacent as per user input and then the depth first search was performed; keeping a track of the visited nodes and those in the open/fringe.

References:

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Second Edition, Pearson Publication
2. Luger, George F. Artificial Intelligence : Structures and strategies for complex problem solving , 2009 ,6th Edition, Pearson Education