

6/5/21

WRITEUP-LAB 4

(18M19CS158)

#> DFS using Time Complexity:-

#include <stdio.h>

#include <time.h>

void dfs (int n, int cost[10][10], int u, int s[])

{

int v;

s[u]=1;

for (v=0; v<n; v++)

{

if ((cost[u][v]==1) && (s[v]==0))

dfs (n, cost, v, s);

}

}

int main()

{
int n, i, j, cost[10][10], s[10], con, lag;

clock_t start, end;

double t;

printf ("Enter the number of nodes\n");

scanf ("%d", &n);

printf ("Enter the adjacency matrix\n");

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

{

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

scanf ("%d", &cost[i][j]);

}

con=0;

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

{

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

s[j]=0;

```

dfs(n, cost, j, s);
lag = 0;
for (i=0; i<n; i++)
{
    if (s[i] == 0)
    {
        lag = 1;
    }
    if (lag == 0)
    {
        con = 1;
    }
    if (con == 1)
    {
        printf("In Graph is connected\n");
    }
    else
    {
        printf("In Graph is not connected\n");
    }
    start = clock();
    dfs(n, cost, j, s);
    end = clock();
    t = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("\n");
    printf("In Time taken by DFS: %Lf\n", t);
    printf("\n");
    return 0;
}

```


#> Tower of Hanoi using time complexity :-

```
#include <time.h>
```

```
#include <stdio.h>
```

```
int TOH(int, char, char, char);
```

```
int main()
```

```
{
```

```
    int n;
```

```
    clock_t t;
```

```
    t = clock();
```

```
    printf("Enter no. of plates:");
```

```
    scanf("%d", &n);
```

```
    int c = TOH(n, 'A', 'C', 'B');
```

```
    printf("\n");
```

```
    printf("Total number of moves = %d\n", c);
```

```
    t = clock() - t;
```

```
    double time_taken = ((double)t) / CLOCKS_PER_SEC;
```

```
    printf("ALGO took %.f seconds to execute\n",
```

```
    time_taken);
```

```
    return 0;
```

```
}
```

```
int TOH(int n, char first, char third, char second)
```

```
{
```

```
    int count;
```

```
    if (n > 0)
```

```
    {
```

```
        count = TOH(n-1, first, second, third);
```

```
        printf("Move disk %d from peg %c to peg %c\n",
```

```
        n, first, third);
```

```
        count++;
```

```
        count += TOH(n-1, second, third, first);
```

```
    }
```

```
    return count;
```

```
}
```