6/5/21

Q -> Tower of Hanoi

```
# include <stdio.h>
# include <time.h>
# include <math.h>
void towers(int n, char s, char t, char d){
    if (n== 1){
        printf("Move disk from %c to %c\n",s,d);
        return;
    }
    towers(n-1, s, d, t);
    printf("Move disk %d from %c to %c \n",n,s,d);
    towers(n-1, t, s, d);
}

int main(){
    int n;
    printf("Enter no of disks : ");
    scanf("%d", &n);
    double toh_time = 0.0;
    clock_t begin = clock();
    towers (n, 'S', 'T', 'D');
    printf("\n Total Steps : %lf ",(pow(2,n)-1));
    clock_t end = clock();
    toh_time+=(double)(end-begin)/clocks_per_sec;
    printf("\n n = %d\t Time = %f \n", n, toh_time);
    return 0;
}
```

d→ DFS Traversal

```
# include < stdio.h>
# include < time.h>
int G[10][10], v[10], n, a[1][10];
void dfs( int i){
    int j;
    printf("\n %d", i);
    v[i] = 1;
    for(j = 0; j<n; j++){
        if(!v[j] && G[i][j] == 1)
            dfs(j);
    }
}
void dfs_c (int n, int G[10][10], int m, int s[]){
    int y;
    s[m] = 1;
    for(y = 0; y < n; y++){
        if((G[m][y] == 1) && (s[y] == 0))
            dfs_c(n, G, y, s);
    }
}
int main(){
    int i, j, con, s[10], flag;
    printf(" Vertices : ");
    scanf("%d", &n);
```

```
printf(" Enter matrix : ");
for(i=0; i<n; i++){
    printf(" Enter row %d :- \n", i+1);
    for(j=0 ; j<n ; j++)
        scanf("%d", &G[i][j]);
}
for(i=0 ; i<n; i++)
        w[i]=0;
printf(" DFS Order :- ");
double tol time = 0.0;
clock t begin = clock();
dfs(0);
con=0;
for(j=0 ; j<n ; j++){
    for(i=0; i<n; i++)
        s[i]=0;
    dfs c(n, G, j, s);
    flag=0;
    for(i=0 ; i<n ; i++){
        if(s[i]==0)
            flag=1;
    }
    if(flag==0)
        con=1;
};
```

```
if (con == 1)
    printf("Graph is Connected");
else { printf("Graph isnt connected");}
clock_t end = clock();
tot_time += (double)(end-begin)/clocks_per_sec;
printf("n = %d \t Time = %f \n", n, tot_time);
return 0;
}
```