Adithya B Shetty
PES1UG19EC009

Program Code:

//dfs

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

double time_elapsed(struct timespec start, struct timespec end)
{
   double t;
   t = (end.tv_sec - start.tv_sec);
   t += (end.tv_nsec - start.tv_nsec) * 0.000000001;
   return t;
}

void dfs(int i,int count, int *visit,int *counter, int **adj,int n)
{
        visit[i]=1;
        counter[i]=count;
        for(int j=0;j<n;j++)
        {
                if(adj[i][j]==1 && visit[j]==0)
                {
                        dfs(j,count,visit,counter,adj,n);
                }
        }
}

int count_components_dfs(int **g, int n)
{
        int count=0;
        int visit[n], counter[n];
        for(int i=0;i<n;i++)
                visit[i]=0;

        for(int i=0;i<n;i++)
        {
                if(visit[i]==0)
                {
                        count++;
                        dfs(i,count,visit,counter,g,n);
                }
        }
        return count;
}
```

```
void main()
{
	int t,i,n,j,temp;
	FILE* fptr=fopen("session8_dfsbfs_ip2.txt","r");
	FILE* fop=fopen("dfs_op2.txt","w");
	fscanf(fptr, "%d", &t);
	struct timespec start, end;

	clock_gettime(CLOCK_REALTIME, &start);
	for(i=0;i<t;i++)
	{
		fscanf(fptr,"%d",&n);
		int **adj=(int **)malloc(n*sizeof(int *));
		for(j=0;j<n;j++)
			adj[j]=(int *)malloc(n*sizeof(int *));
		for(j=0;j<n;j++)
			for(int k=0;k<n;k++)
			{
				fscanf(fptr,"%d",&adj[j][k]);
			}
		temp=count_components_dfs(adj,n);
		fprintf(fop,"%d\n",temp);
	}
	clock_gettime(CLOCK_REALTIME, &end);
	fprintf(fop,"%lf sec spent on dfs\n",time_elapsed(start, end));
	fclose(fptr);
	fclose(fop);
}
```

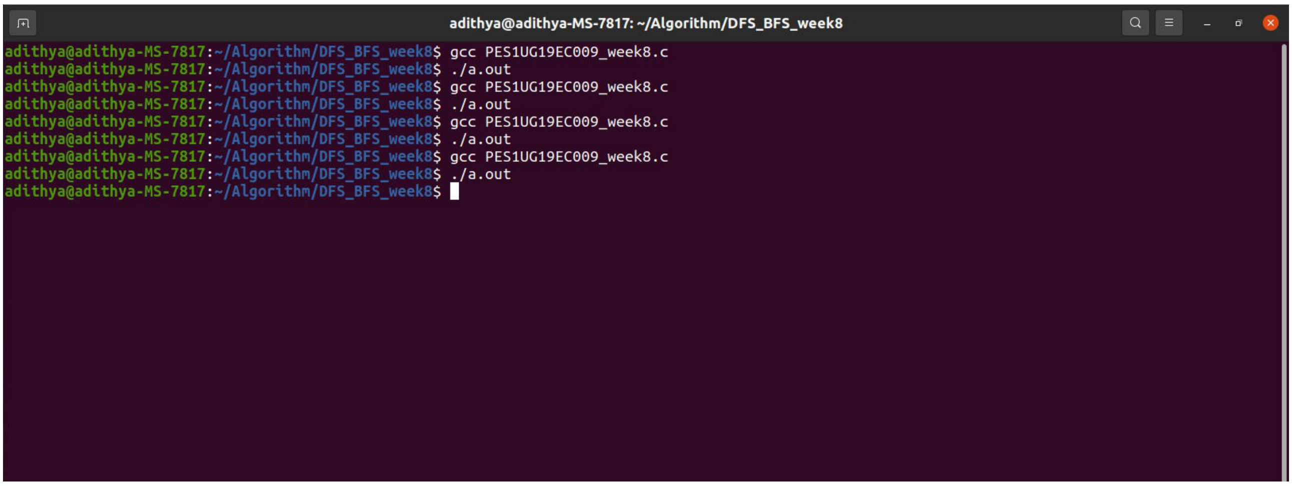
//bfs

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

double time_elapsed(struct timespec start, struct timespec end)
{
   double t;
   t = (end.tv_sec - start.tv_sec);
   t += (end.tv_nsec - start.tv_nsec) * 0.000000001;
   return t;
}

void bfs(int i, int *visit, int **adj,int n,int* queue,int f,int r)
{
	visit[i]=1;
	for(int j=0;j<n;j++)
	{
		if((adj[i][j]==1) && (visit[j]==0))
		{
```

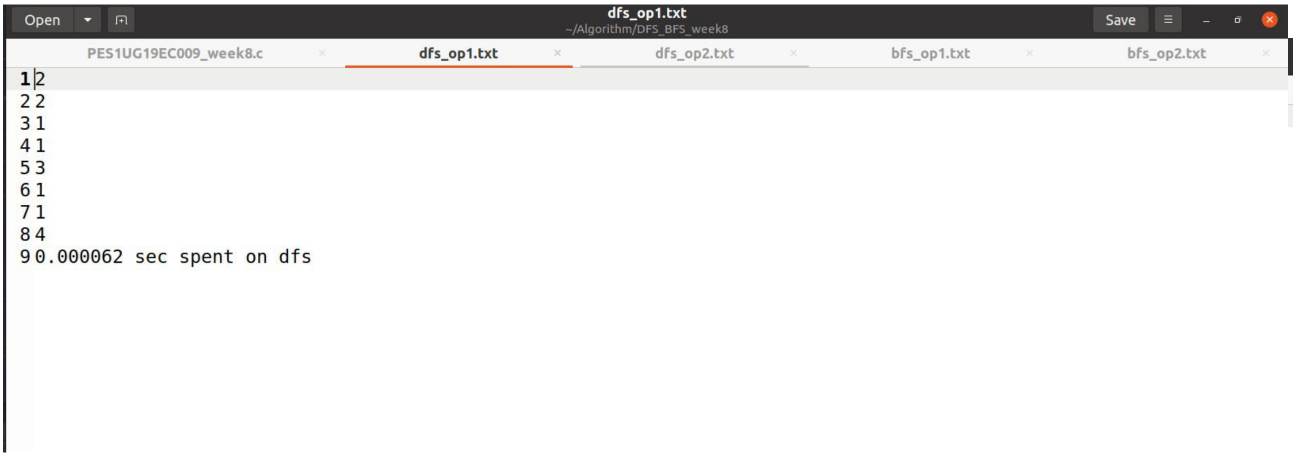
```
                visit[j]=1;
                queue[++r]=j;

            }
        }

}

int count_components_bfs(int **g, int n)
{
        int count=0,queue[n],f=-1,r=-1;;
        int visit[n];
        for(int i=0;i<n;i++)
            visit[i]=0;
        for(int i=0;i<n;i++)
        {
            if(visit[i]==0)
            {
                ++count;
                bfs(i,visit,g,n,queue,f,r);
            }
        }
        while(f<r)
        {
            bfs(queue[f],visit,g,n,queue,f,r);
        }
        return count;
}

void main()
{
        int t,i,n,j,temp;
        FILE* fptr=fopen("session8_dfsbfs_ip2.txt","r");
        FILE* fop=fopen("bfs_op2.txt","w");
      fscanf(fptr, "%d", &t);
      struct timespec start, end;
        clock_gettime(CLOCK_REALTIME, &start);
        for(i=0;i<t;i++)
        {
            fscanf(fptr,"%d",&n);
            int **adj=(int **)malloc(n*sizeof(int *));
            for(j=0;j<n;j++)
                adj[j]=(int *)malloc(n*sizeof(int *));
            for(j=0;j<n;j++)
                for(int k=0;k<n;k++)
                {
                    fscanf(fptr,"%d",&adj[j][k]);
                }
            temp=count_components_bfs(adj,n);
            fprintf(fop,"%d\n",temp);
        }
        clock_gettime(CLOCK_REALTIME, &end);
```

```
	fprintf(fop,"%lf sec spent on bfs\n",time_elapsed(start, end));
	fclose(fptr);
	fclose(fop);
}
```

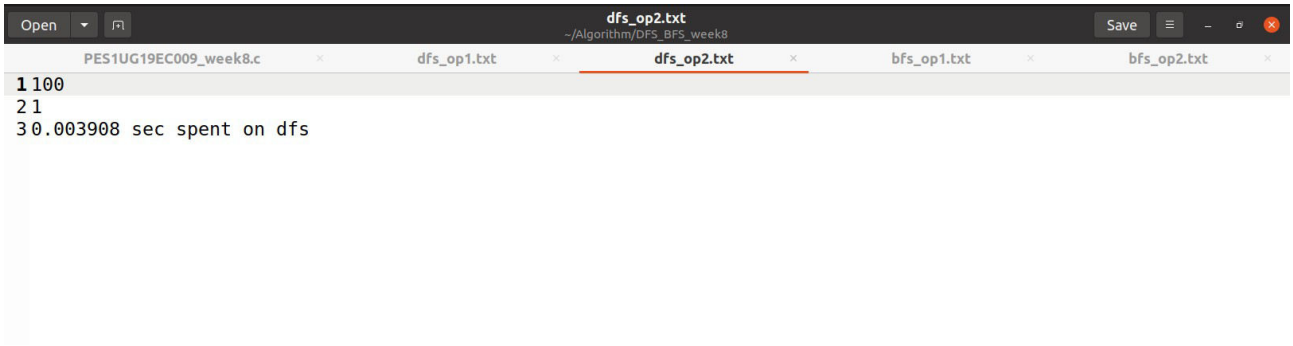
OUTPUT



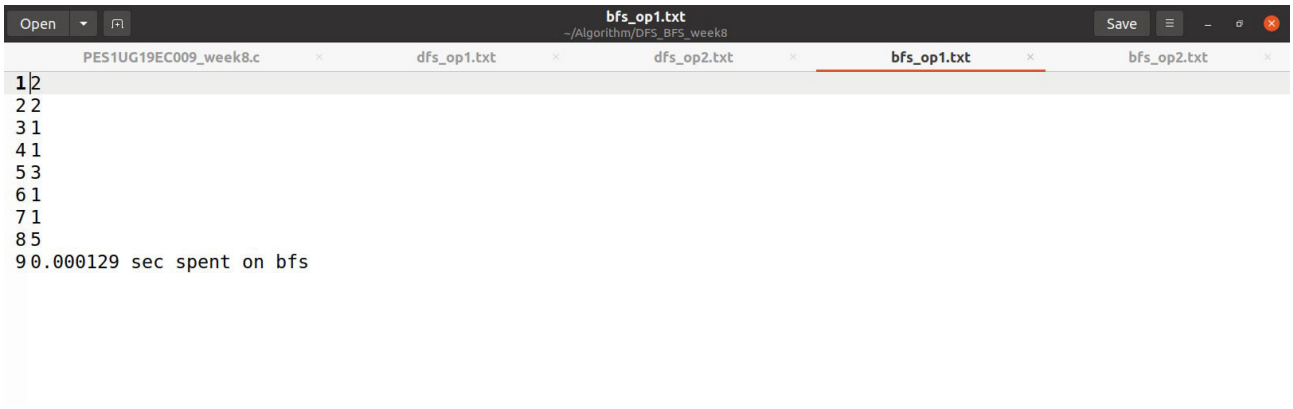
DFS OUTPUT 1



DFS OUTPUT 2



BFS OUTPUT 1



BFS OUTPUT 2