DAA Lab - Session 8 - DFS and BFS

Decrease-and-Conquer: Implementation of DFS and BFS algorithms

1. Find the number of components in the given undirected graph using **DFS and then BFS algorithms.**

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define Max 200

double time\_elapsed(struct timespec start, struct timespec end);

int graph[Max][Max],visited[Max],q[Max];

int f=-1, r=-1;

void initgraph(int graph[][Max],int n){

int i,j;

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

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

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

}

void dfs(int v , int graph[][Max],int visited[],int n,int label)

{

int w;

visited[v]=label;

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

{

if((graph[v][w]==1)&&(visited[w]==0))

dfs(w,graph,visited,n,label);

}

}

int qisempty()

{

if((f==-1&&r==-1)||(f>r))

return 1;

return 0;

}

void qinsert(int v)

{

if(f==-1) f++;

q[++r] = v;

}

int qdelete()

{

int v;

if ( f==-1) return 0;

v= q[f++];

return v;

}

void bfs(int v , int graph[][Max],int visited[],int n,int label)

{

int w;

visited[v]=1;

qinsert(v);

while(!qisempty())

{ v = qdelete();

for(w=1;w<=n;w++)

{

if ( graph[v][w]==1 && visited[w]==0)

{ qinsert(w);

visited[w]=1;

}

}

}

}

int count\_components\_dfs(int graph[][Max],int n)

{

int i;

int label=0;

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

visited[i]=0;

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

{

if(visited[i]==0)

{

++label;

dfs(i,graph,visited,n,label);

}

}

return label;

}

int count\_components\_bfs(int graph[][Max],int n)

{

int i;

int label=0;

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

visited[i]=0;

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

{

if(visited[i]==0)

{

++label;

bfs(i,graph,visited,n,label);

}

}

return label;

}

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;

}

int main(){

int n,i,num;char temp;

struct timespec start, end;

//printf("The Number of Connected Components in DFS for given inputs:\n");

printf("The Number of Connected Components in BFS for given inputs:\n");

int count;

clock\_gettime(CLOCK\_REALTIME, &start);

scanf("%d",&num);

for(int j=0;j<num;j++)

{

scanf("%c",&temp);

scanf("%d",&n);

initgraph(graph,n);

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

visited[i]=0;

}

//int numberOfComponents1 = count\_components\_dfs(graph,n);

//printf("%d\n",numberOfComponents1);

int numberOfComponents2 = count\_components\_bfs(graph,n);

printf("%d\n",numberOfComponents2);

}

clock\_gettime(CLOCK\_REALTIME, &end);

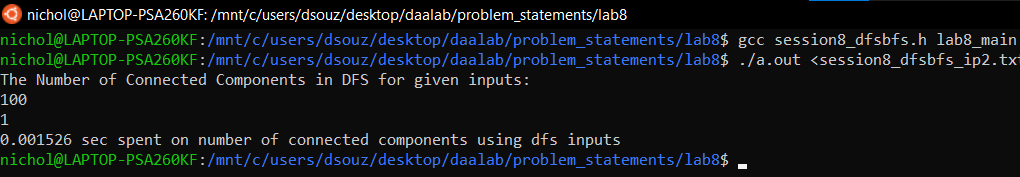
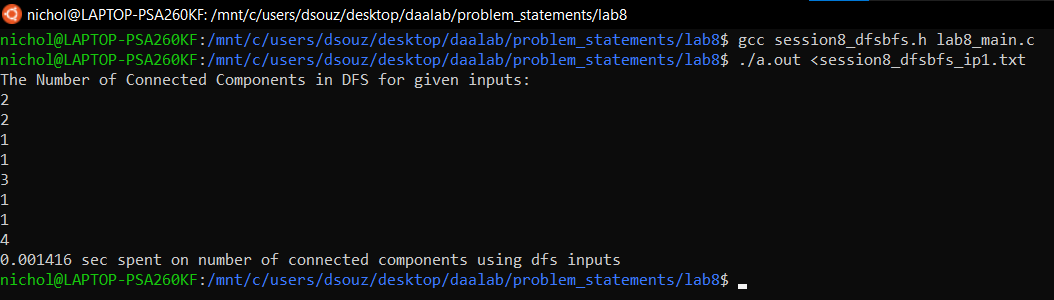
//printf("%lf sec spent on number of connected components using dfs inputs\n", time\_elapsed(start, end));

printf("%lf sec spent on number of connected components using bfs inputs\n", time\_elapsed(start, end));

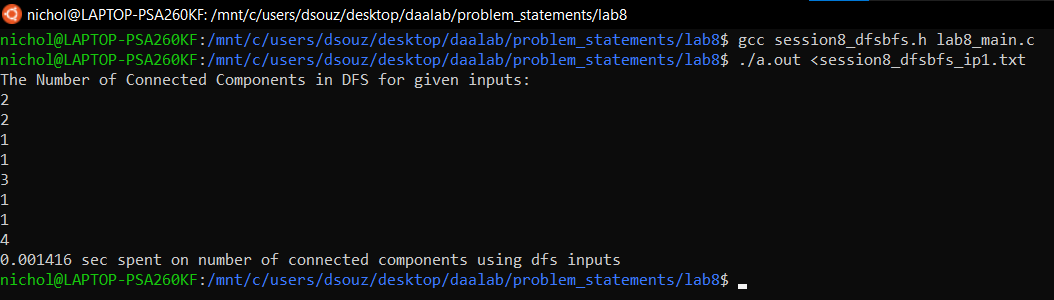
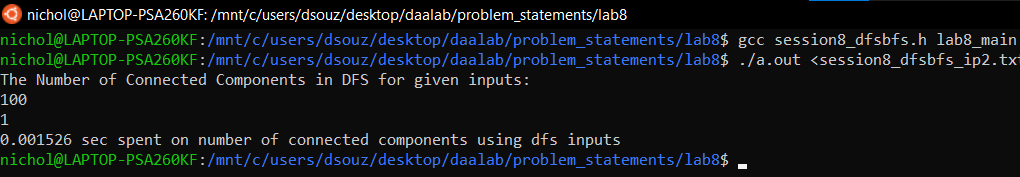
}

**Output snippets:**

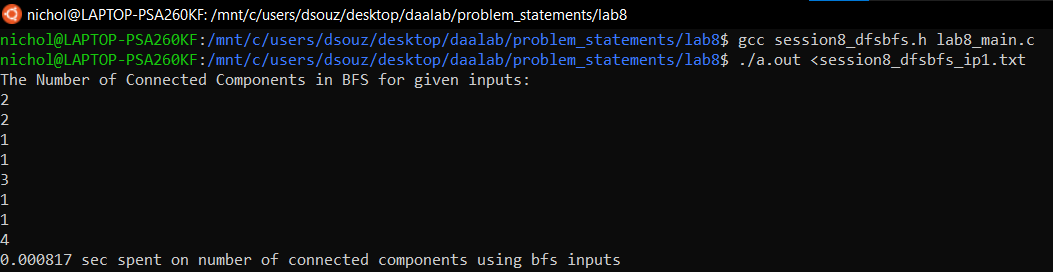
**DFS input 1**



**DFS input 2**



**BFS input 1**



**BFS input 2**

