Пензенский государственный университет

Кафедра "Вычислительной техники"

**Отчёт**

По лабораторной работе №3

По дисциплине: “[Логика и основы алгоритмизации в инженерных задачах](http://moodle.pnzgu.ru/course/view.php?id=48560)”.

На тему «Унарные и бинарные операции над графами.»

***Выполнили студенты группы 19вв1:***

*Кочетков К.*

***Приняли:***

*Митрохин М.А.*

*Юрова О.А.*

Пенза 2020

**Общие сведения.**

Все унарные операции над графами можно объединить в две группы. Первую группу составляют операции, с помощью которых из исходного графа *G*1*,* можно построить граф *G*2 с меньшим числом элементов. В группу входят операции удаления ребра или вершины, отождествления вершин, стягивание ребра. Вторую группу составляют операции, позволяющие строить графы с большим числом элементов. В группу входят операции расщепления вершин, добавления ребра.

*Отождествление вершин.* В графе *G*1 выделяются вершины *и,v.* Определяют окружение *Q*1 вершины *u*,и окружение *Q*2 вершины *v,* вычисляют их объединение *Q* = *Q1* * Q2.* Затем над графом *G*1 выполняются следующие преобразования:

* из графа *G*1 удаляют вершины *u,* *v (H*1 *= G*1 *- u - v);*
* к графу *Н*1присоединяют новую вершину *z (H*1 *= H*1 *+z);*
* вершину *z* соединяют ребром с каждой из вершин *w*1*Q*

*(G*2 *= H*1 *+ zwi*, *i =* 1,2,3*,…).*

*Стягивание ребра.* Данная операция является операцией отождествления смежных вершин *и, v* в графе *G*1.

Наиболее важными бинарными операциями являются: объединение, пересечение, декартово произведение и кольцевая сумма.

*Объединение.* Граф *G* называется объединением или наложением графов *G*1 и *G*2, если *VG = V*1*V*2*; UG = U*1* U*2 (рис. 1).

**U**

*v*1

*v*2

*v*3

*v*4

*v*3

*v*4

*v*5

*v*2

*v*1

*v*3

*v*4

*v*5

Рис. 1. Объединение графов *G*1, *G*2

Объединение графов *G*1 и *G*2 называется дизъюнктным, если *V*1*V*2 *= *. При дизъюнктном объединении никакие два из объединяемых графов не должны иметь общих вершин.

*Пересечение.* Граф *G* называется пересечением графов *G*1, *G*2,если *VG = V*1*V*2и *UG = U*1*U*2 (риc.2). Операция "пересечения" записывается следующим образом: *G = G*1*G*2*.*

**∩**

*v*1

*v*2

*v*3

*v*5

*v*3

*v*4

*v*6

*v*2

*v*1

*v*6

*v*4

*v*5

*v*1

*v*4

*v*6

*v*5

*v*3

*v*2

Рис.2. Пересечение графов *G*1, *G*2*.*

*Декартово произведение.* Граф *G* называется декартовым произведением графов *G*1 и *G*2 если *VG* = *V*1*V*2 —декартово произведение множеств вершин графов *G*1, *G*2, а множество ребер *U*c задается следующим образом: вершины (*zi*, *vk*) и (*zj*, *vl*) смежны в графе *G* тогда и только тогда, когда *zi* = *zj*(*i* = *j*), a *v*k и *vl* смежны в *G*2 или *vk* = *vl*(*k* = *l*), смежны в графе *G*1 (см. рис.3).

**X**

*z*1

*z*2

*v*1

*v*3

*v*2

*z*1*v*1

*z*1*v*2

*z*1*v*3

*z*2*v*1

*z*2*v*2

*z*2*v*3

Рис. 3. Декартово произведение графов *G*1, *G*2

*Кольцевая сумма* графов представляет граф, который не имеет изолированных вершин и состоит из ребер, присутствующих либо в первом исходном графе, либо во втором. Кольцевая сумма определяется следующим соотношением: *G* = *G*1  *G*2 (рис.4).

**⊕**

*v*1

*v*2

*v*3

*v*5

*v*3

*v*4

*v*6

*v*2

*v*1

*v*4

*v*5

*v*1

*v*4

*v*6

*v*5

*v*3

*v*2

Рис.4. Кольцевая сумма графов *G1, G2*

**Листинг:**

#include <iostream>

#include <stdio.h>

#include <locale>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

int main()

{

system("chcp 1251");

system("cls");

srand(time(NULL));

int m,n,z,z1,z2,z3;

int\*\* arr;

int\*\* arr4;

int\*\* arr5;

int\*\* arr6;

int\*\* arr7;

printf("Введите количество вершин ");

scanf("%d",&m);

z=m-1;

z1=z;

z2=m+1;

z3=z2;

printf("Матрица M1:\n ");

for(int i=1;i<m+1;i++){printf("%d ",i);}

printf("\n-----------\n");

n=m;

arr = (int\*\*)malloc(n \* sizeof(int));

arr5 = (int\*\*)malloc(n \* sizeof(int));

arr4 = (int\*\*)malloc(z2 \* sizeof(int));

for(int i=0;i<m+1;i++){

if(i<m){arr[i]=(int\*)malloc(m \* sizeof(int));

arr5[i]=(int\*)malloc(m \* sizeof(int));}

arr4[i]=(int\*)malloc(z3 \* sizeof(int));

for(int j=0;j<m;j++){

if(i<m){arr[i][j]=rand()%2;

if(i==j){arr[i][j]=0;}

if(j<i){arr[i][j]=arr[j][i];}

printf(" %d",arr[i][j]);

arr4[i][j]=arr[i][j];arr5[i][j]=arr[i][j];}

}

printf("\n");

}

int\*\* arr1;

printf("Матрица M2:\n");

n=m;

arr1 = (int\*\*)malloc(n \* sizeof(int));

arr6 = (int\*\*)malloc(n \* sizeof(int));

for(int i=0;i<m;i++){

arr1[i]=(int\*)malloc(m \* sizeof(int));

arr6[i]=(int\*)malloc(m \* sizeof(int));

for(int j=0;j<m;j++){

arr1[i][j]=rand()%2;

if(i==j){arr1[i][j]=0;}

if(j<i){arr1[i][j]=arr1[j][i];}

printf(" %d",arr1[i][j]);

arr6[i][j]=arr1[i][j];

}

printf("\n");

}

int h,k,h1;

for(int i=0;i<1;){

printf("Введите 2 вершины которые хотие отождествить:");

scanf("%d %d",&h,&k);

if(k==h||h>m||k>m||k==0||h==0){printf("Вы ввели вершины неправильно. Попробуйте ещё раз\n");}

else i++;

}

if(k<h){h1=k;k=h;h=h1;}

h--;

k--;

int\*\* arr2;

printf("Отождествлённая матрица:\n");

arr[h][k]=0;

arr[k][k]=0;

arr[h][h]=0;

arr[k][h]=0;

for(int i=0;i<m;i++)

{

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

if(h==i){

if(arr[h][j]==1||arr[k][j]==1){arr[h][j]=1;arr[k][j]=0;}

}

if (h==j){

if(arr[i][h]==1||arr[i][k]==1){arr[i][h]=1;arr[i][k]=0;}

}

}

}

printf("\n");

arr2 = (int\*\*)malloc(z \* sizeof(int));

for(int i=0;i<z;i++){

arr2[i]=(int\*)malloc(z1 \* sizeof(int));

for(int j=0,p;j<z1;j++){

p=i;

if(i>=k){p=i+1;}

if(j<k){arr2[i][j]=arr[p][j];}

if(j>=k){arr2[i][j]=arr[p][j+1];}

printf(" %d", arr2[i][j]);

}

printf("\n");

}

for(int i=0;i<1;){

printf("Введите ребро которое хотите стянуть:");

scanf("%d %d",&h,&k);

if((k>m||h>m||k<0||h<0)||(arr1[h-1][k-1]!=1)||(h==k)){printf("Вы ввели вершины неправильно или такой вершины нет. Попробуйте ещё раз\n");}

else i++;

}

if(k<h){h1=k;k=h;h=h1;}

h--;

k--;

printf("Отождествлённая матрица:\n");

arr1[h][k]=0;

arr1[k][k]=0;

arr1[h][h]=0;

arr1[k][h]=0;

for(int i=0;i<m;i++)

{

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

if(h==i){

if(arr1[h][j]==1||arr1[k][j]==1){arr1[h][j]=1;arr1[k][j]=0;}

}

if (h==j){

if(arr1[i][h]==1||arr1[i][k]==1){arr1[i][h]=1;arr1[i][k]=0;}

}

}

}

printf("\n");

arr2 = (int\*\*)malloc(z \* sizeof(int));

for(int i=0;i<z;i++){

arr2[i]=(int\*)malloc(z1 \* sizeof(int));

for(int j=0,p;j<z1;j++){

p=i;

if(i>=k){p=i+1;}

if(j<k){arr2[i][j]=arr1[p][j];}

if(j>=k){arr2[i][j]=arr1[p][j+1];}

printf(" %d", arr2[i][j]);

}

printf("\n");

}

for(int i=0;i<1;){

printf("Введите вершину которую хотите расщепить");

scanf("%d",&k);

if(k<=0||k>n){printf("Вы ввели вершину которой нет, введите ещё раз:\n");}

else i++;

}

k--;

for(int i=0;i<z2;i++){

for(int j=0;j<z2;j++){

if(i==z2-1||j==z2-1){

if(i==j){arr4[i][j]=0;}

if(i==z2-1&&arr4[i][j]!=0){arr4[i][j]=arr4[k][j];}

if(j==z2-1&&arr4[i][j]!=0){arr4[i][j]=arr4[i][k];}

}

printf(" %d", arr4[i][j]);

}

printf("\n");

}

printf("Объединение матриц:\n");

arr7 = (int\*\*)malloc(n \* sizeof(int));

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

arr7[i]=(int\*)malloc(m \* sizeof(int));

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

if(arr5[i][j]==1||arr6[i][j]==1){arr7[i][j]=1;}

else arr7[i][j]=0;

printf(" %d", arr7[i][j]);

}

printf("\n");

}

printf("Пересечение матриц:\n");

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

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

if(arr5[i][j]==1&&arr6[i][j]==1){arr7[i][j]=1;}

else arr7[i][j]=0;

printf(" %d", arr7[i][j]);

}

printf("\n");

}

printf("Кольцевая сумма:\n");

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

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

if((arr5[i][j]==1&&arr6[i][j]==0)||(arr5[i][j]==0&&arr6[i][j]==1)){arr7[i][j]=1;}

if(arr5[i][j]==arr6[i][j]){arr7[i][j]=0;}

printf(" %d", arr7[i][j]);

}

printf("\n");

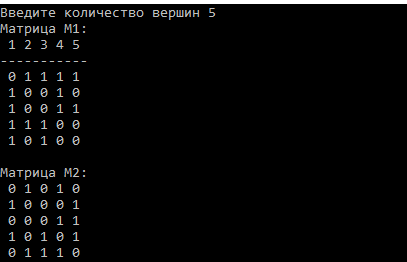
}

system("pause");

}

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) две матрицы *M*1*, М*2 смежности неориентированных помеченных графов *G*1, *G*2. Выведите сгенерированные матрицы на экран.



### **Задание 2**

1. Для матричной формы представления графов выполните операцию:

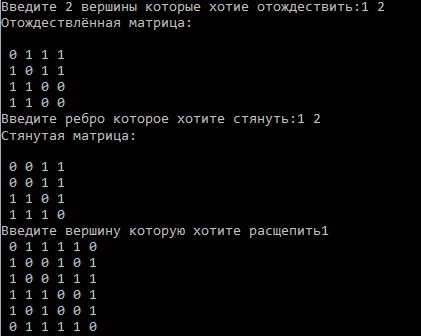
а) отождествления вершин

б) стягивания ребра

в) расщепления вершины

Номера выбираемых для выполнения операции вершин ввести с клавиатуры.

Результат выполнения операции выведите на экран.



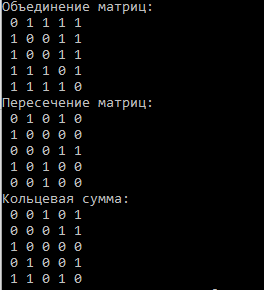
**Задание 3**

1. Для матричной формы представления графов выполните операцию:

а) объединения *G* = *G*1  *G*2

б) пересечения *G* = *G*1  *G*2

в) кольцевой суммы *G* = *G*1  *G*2



**Вывод:** в ходе данной работы я научился выполнять унарные и бинарные действия над графами.