**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

дисциплина: Дискретная математика

Лабораторная работа № 4.2

тема: «Циклы»

|  |  |
| --- | --- |
|  | Выполнил: ст. группы ВТ-22  Макаров Даниил Сергеевич  Проверил: Рязанов Ю.Д. |

Белгород

2018г.

**Цель занятия:** изучить разновидности циклов в графах, научиться генерировать случайные графы, определять их принадлежность к множеству эйлеровых и гамильтоновых графов, находить все эйлеровы и гамильтоновы циклы в графах.

**Задания**

1. Разработать и реализовать алгоритм генерации случайного графа, содержащего n вершин и m ребер.

2. Написать программу, которая:

а) в течение десяти секунд генерирует случайные графы, содержащие n вершин и m ребер;

б) для каждого полученного графа определяет, является ли он эйлеровым или гамильтоновым;

в) подсчитывает общее количество сгенерированных графов и количество графов каждого типа.

Результат работы программы представить в виде таблицы

3. Выполнить программу при n = 8, 9, 10 и сделать выводы.

4. Привести пример диаграммы графа, который является эйлеровым, но не гамильтоновым. Найти в нем все эйлеровы циклы.

5. Привести пример диаграммы графа, который является гамильтоновым, но не эйлеровым. Найти в нем все гамильтоновы циклы.

6. Привести пример диаграммы графа, который является эйлеровым и гамильтоновым. Найти в нем все эйлеровы и гамильтоновы циклы.

7. Привести пример диаграммы графа, который не является ни эйлеровым, ни гамильтоновым.

|  |  |  |  |
| --- | --- | --- | --- |
| **N = 8** |  |  |  |
| m | Эйлеровы графы | Гамильтоновы | Все |
| 8 | 209 | 12 | 7815 |
| 9 | 0 | 2348 | 325585 |
| 10 | 4947 | 8497 | 241335 |
| 11 | 0 | 18098 | 165991 |
| 12 | 2377 | 34410 | 139070 |
| 13 | 0 | 51979 | 119459 |
| 14 | 2113 | 84271 | 135518 |
| 15 | 0 | 103695 | 134023 |
| 16 | 2192 | 138929 | 158209 |
| 17 | 0 | 171617 | 182691 |
| 18 | 2473 | 202698 | 208500 |
| 19 | 0 | 250566 | 253422 |
| 20 | 2953 | 282080 | 283167 |
| 21 | 0 | 307145 | 307441 |
| 22 | 2490 | 310550 | 310597 |
| 23 | 0 | 253389 | 253389 |
| 24 | 1504 | 294369 | 294369 |
| 25 | 0 | 270981 | 270981 |
| 26 | 0 | 246621 | 246621 |
| 27 | 0 | 226896 | 226896 |
| 28 | 0 | 179254 | 179254 |

|  |  |  |  |
| --- | --- | --- | --- |
| **N=9** |  |  |  |
| m | Эйлеровы графы | Гамильтоновы | Все |
| 9 | 0 | 47 | 228626 |
| 10 | 4123 | 609 | 303943 |
| 11 | 0 | 2378 | 217270 |
| 12 | 2050 | 7627 | 185009 |
| 13 | 0 | 14629 | 137254 |
| 14 | 1094 | 24702 | 110280 |
| 15 | 0 | 32501 | 87309 |
| 16 | 762 | 46489 | 87857 |
| 17 | 0 | 58205 | 86548 |
| 18 | 658 | 69730 | 88522 |
| 19 | 0 | 83412 | 96147 |
| 20 | 726 | 99055 | 107547 |
| 21 | 0 | 122701 | 128308 |
| 22 | 929 | 145232 | 148773 |
| 23 | 0 | 171515 | 173759 |
| 24 | 1150 | 206496 | 207703 |
| 25 | 0 | 240356 | 240926 |
| 26 | 1363 | 262045 | 262309 |
| 27 | 0 | 284338 | 284418 |
| 28 | 1191 | 293617 | 293636 |
| 29 | 0 | 290806 | 290811 |
| 30 | 876 | 281666 | 281666 |
| 31 | 0 | 279290 | 279290 |
| 32 | 447 | 265869 | 265869 |
| 33 | 0 | 249446 | 249446 |
| 34 | 0 | 233100 | 233100 |
| 35 | 0 | 212803 | 212803 |
| 36 | 0 | 176153 | 176153 |

|  |  |  |  |
| --- | --- | --- | --- |
| **N = 10** |  |  |  |
| m | Эйлеровы графы | Гамильтоновы | Все |
| 10 | 2411 | 18 | 258513 |
| 11 | 0 | 188 | 287367 |
| 12 | 1570 | 812 | 215654 |
| 13 | 0 | 2359 | 159501 |
| 14 | 740 | 5154 | 118351 |
| 15 | 0 | 9448 | 89475 |
| 16 | 389 | 13903 | 69759 |
| 17 | 0 | 18461 | 57492 |
| 18 | 215 | 23100 | 50223 |
| 19 | 0 | 26926 | 45743 |
| 20 | 206 | 31068 | 44145 |
| 21 | 0 | 34840 | 44062 |
| 22 | 183 | 39210 | 45746 |
| 23 | 0 | 44007 | 48663 |
| 24 | 188 | 50920 | 54254 |
| 25 | 0 | 57556 | 59965 |
| 26 | 251 | 69372 | 71041 |
| 27 | 0 | 81058 | 82212 |
| 28 | 289 | 98999 | 99784 |
| 29 | 0 | 123609 | 124106 |
| 30 | 408 | 147051 | 147360 |
| 31 | 0 | 165204 | 165422 |
| 32 | 509 | 201426 | 201519 |
| 33 | 0 | 226752 | 226788 |
| 34 | 542 | 237018 | 237033 |
| 35 | 0 | 242713 | 242719 |
| 36 | 499 | 246234 | 246236 |
| 37 | 0 | 243741 | 243741 |
| 38 | 334 | 240279 | 240279 |
| 39 | 0 | 230300 | 230300 |
| 40 | 177 | 223596 | 223596 |
| 41 | 0 | 211401 | 211401 |
| 42 | 0 | 195636 | 195636 |
| 43 | 0 | 188261 | 188261 |
| 44 | 0 | 171703 | 171703 |
| 45 | 0 | 142365 | 142365 |

График функции, описывающий отношение количества эйлеровых/гамильтоновых графов к общему количеству графов, имеет вид

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

//Выделяет память для матрицы

int \*\*getMem(int n){

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

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

r[i] = malloc(sizeof(int)\*n);

}

return r;

}

void freeMem(int \*\*G, int n){

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

free(G[i]);

}

free(G);

}

void fillGraf (int \*\*a, int n, int m){

unsigned i, j, k=0;

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

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

a[i][j] = 0;

}

}

while (k<m){

do{

i=rand()%n;

j=rand()%n;

}while(a[i][j]&&a[j][i]);

if (i!=j){

a[i][j]=a[j][i]=1;

k++;

}

}

}

int searchHamiltonianCycle(int v, int \*\*G, int n, int \*visited, int w, int d) {

//Если все вершины были посещены и сущ-т ребро, соед. 1 с посл.

if (d == 1 && G[w][v]) return 1;

visited[v] = 1;

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

//Для всех смежных в.

if (G[t][v]){

if (!visited[t]){

if (searchHamiltonianCycle(t, G, n, visited, w, d-1)){

return 1;

}

}

}

}

visited[v] = 0;

return 0;

}

int isEulerGraph(int \*\*G, int n){

unsigned k;

//Для каждой из вершин определим ее степень k

//Если степени всех вершин четны, граф Эйлеров

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

k = 0;

for (int j = i+1; j<n; j++){

if (G[i][j]){

k++;

}

}

if (k % 2){

return 0;

}

}

return 1;

}

int main(){

int n;

printf("Input n: ");

scanf("%i", &n);

long int starttime;

int C = n\*(n-1)/2; //=C\_n^2

int euler, hamil, all;

int \*\*G = getMem(n);

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

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

euler = hamil = all = 0;

starttime = time(NULL);

while (time(NULL)<starttime+1){

fillGraf(G, n, i);

if (isEulerGraph(G, n)){

euler++;

}

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

visited[j] = 0;

}

if (searchHamiltonianCycle(0, G, n, visited, 0, n)){

hamil++;

}

all++;

}

printf("m = %i, Euler: %i, Hamilton: %i, All: %i\n", i, euler, hamil, all);

}

return 0;

}