**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

Лабораторная работа № 3.2

дисциплина: Дискретная математика

тема: «Транзитивное замыкание отношения»

|  |  |
| --- | --- |
|  | Выполнил: ст. группы ПВ-21  Браткова Ирина Олеговна  Проверил: Рязанов Ю.Д. |

Белгород 2017

**Цель занятия**: изучить и выполнить сравнительный анализ алгоритмов вычисления транзитивного замыкания отношения.

***Задания***

1. Изучить и программно реализовать алгоритмы объединения степеней и Уоршалла для вычисления транзитивного замыкания отношения.
2. Разработать и программно реализовать генератор отношений на множестве мощности N и содержащих заданное число пар.
3. Разработать и написать программу, которая генерирует 1000 отношений на множестве мощности N с заданным числом пар, для каждого отношения вычисляет транзитивное замыкание двумя алгоритмами и определяет время выполнения каждого алгоритма. Время вычисления транзитивного замыкания различных отношений на множестве мощности N с заданным числом пар может быть разным, поэтому программа так же должна определять минимальное и максимальное время вычисления транзитивного замыкания сгенерированных отношений. Выполнить программу при N = 50, 100 и 150. Результат для каждого N представить в виде таблицы.

**Задание 1**: Изучить и программно реализовать алгоритмы объединения степеней и Уоршалла для вычисления транзитивного замыкания отношения.

void tranzit\_uol (int a[N][N], int c[N][N])

***//алгоримт транзитивного замыкания Уоршалла***

{

copy\_ot(a, c); int x, y, z;

for (z=0; z<N; z++)

for (y=0; y<N; y++)

for (x=0; x<N; x++)

{ c[x][y]=c[x][y]+c[x][z]\*c[z][y];

if (c[x][y] > 1) c[x][y]=1; }

}

void tranzit\_step (int a[N][N], int c[N][N])

***//Алгоритм объединения степеней***

{

int t1[N][N], t2[N][N], i;

copy\_ot(t1, a); copy\_ot(c, a);

for (i=2; i<=N; i++)

{

k += comp(t1, a, t2); ***//композиция***

ob(c, t2, c); ***//обединение*** copy\_ot(t1, t2);

}

}

|  |  |
| --- | --- |
| void copy\_ot (int a[N][N], int b[N][N])  ***//Копирует отношение a мощности N в b той же мощности***  {  int i, j;  for (i=0; i<N; i++)  for (j=0; j<N; j++) b[i][j]=a[i][j];  } | void null\_matr (int s[N][N])  **/*/Обнуление элементов матрицы***  {  int i=0, j=0;  for (i=0; i<N; i++)  for (j=0; j<N; j++) s[i][j]=0;  } |

void output\_matr(int a[N][N])

***// вывод матрицы***

{ int i, j; for (i=0;i<N;i++)

{ for (j=0;j<N;j++) { printf("%d ", a[i][j]); }

printf("\n"); } printf("\n");

}

**Задание 2:** Разработать и программно реализовать генератор отношений на множестве мощности N и содержащих заданное число пар.

void generation (int a[N][N], unsigned count1, int n)

***// Генерация отношения a на множестве мощности N, содержащем count1 пар, где n – число, по которому будет генерироваться случайно число функцией rand.***

{

srand(n); int i,j,k;

for (k=0; k<count1; ++k)

{

do

{

i=rand()%N; j=rand()%N;

}

while (a[i][j]); a[i][j]=1;

}

}

**Задание 3**: Разработать и написать программу, которая генерирует 1000 отношений на множестве мощности N с заданным числом пар..

void timer\_1 (int a[N][N], int kolpar)

***//Генерирует 1000 отношений на множестве а мощности N с заданным числом пар kolpar, выводит на экран минимальное и максимальное время выполнения транзитивного замыкания из 1000 отношений данного множества для алгоритма Уоршалла.***

{

int i=0, c[N][N]; double min, max, tmp1, tmp2;

null\_matr(a); null\_matr(c); generation(a, kolpar, i);

clock\_t time\_start= clock(); tranzit\_uol(a, c); clock\_t time\_end = clock() - time\_start;

min = (double)time\_end/CLOCKS\_PER\_SEC;

max = (double)time\_end/CLOCKS\_PER\_SEC;

for (i=1; i<OTN; i++)

{

null\_matr(a); null\_matr(c);

generation(a, kolpar, i);

clock\_t time\_start= clock(); tranzit\_uol(a, c); clock\_t time\_end = clock() - time\_start;

tmp1 = (double)time\_end/CLOCKS\_PER\_SEC;

if ((tmp1 - max) > EPS) max = tmp1;

if ((tmp1 - min) < EPS) min = tmp1;

}

printf("Уоршалл. При N = %d, min = %f, max = %f\n", N, min, max);

}

void timer\_2 (int a[N][N], int kolpar)

{

***//Генерирует 1000 отношений на множестве а мощности N с заданным числом пар kolpar, выводит на экран минимальное и максимальное время выполнения транзитивного замыкания из 1000 отношений данного множества для объединения степеней.***

int i=0, c[N][N]; double min, max, tmp1, tmp2;

null\_matr(a); null\_matr(c); generation(a, kolpar, i);

clock\_t time\_start= clock(); tranzit\_uol(a, c); clock\_t time\_end = clock() - time\_start;

min = (double)time\_end/CLOCKS\_PER\_SEC;

max = (double)time\_end/CLOCKS\_PER\_SEC;

for (i=1; i<OTN; i++)

{

null\_matr(a); null\_matr(c); generation(a, kolpar, i);

clock\_t time\_start= clock(); tranzit\_step(a, c); clock\_t time\_end = clock() - time\_start;

tmp1 = (double)time\_end/CLOCKS\_PER\_SEC;

if ((tmp1 - max) > EPS) max = tmp1;

if ((tmp1 - min) < EPS) min = tmp1;

}

printf("Степень. При N = %d, min = %f, max = %f\n", N, min, max);

}

***Вызов функций в главной функции:***

void main()

{

int a[N][N]; null\_matr(a); int n = 1; timer\_2(a, n); timer\_1(a, n); null\_matr(a);

}

N = 50

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число пар в отношении | 1 | | N2/4 | | N2/2 | |  | | N2 | |
| min | max | min | max | min | max | min | max | min | max |
| Объединение степеней | 0,025 | 0,025 | 0,028 | 0,028 | 0,027 | 0,027 | 0,027 | 0,027 | 0,031 | 0,031 |
| Алг. Уоршалла | 0 | 0,003 | 0 | 0,017 | 0 | 0,016 | 0 | 0,017 | 0 | 0.016 |

N = 25

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число пар в отношении | 1 | | N2/4 | | N2/2 | |  | | N2 | |
| min | max | min | max | min | max | min | max | min | max |
| Объед. степеней | 0.005 | 0.007 | 0.004 | 0.009 | 0.005 | 0.008 | 0.001 | 0.003 | 0.001 | 0.006 |
| Алг. Уоршалла | 0 | 0,005 | 0 | 0,006 | 0 | 0,006 | 0 | 0,006 | 0 | 0,005 |

N = 80

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число пар в отношении | 1 | | N2/4 | | N2/2 | |  | | N2 | |
| min | max | min | max | min | max | min | max | min | max |
| Объед. степеней | 0,13 | 0.178 | 0.124 | 0.139 | 0,125 | 0.138 | 0.121 | 0,133 | 0.119 | 0.132 |
| Алг. Уоршалла | 0,001 | 0,005 | 0,002 | 0,007 | 0,001 | 0,009 | 0,001 | 0,021 | 0,001 | 0,022 |