Kpi-best

МІНІСТЕРСТВО НАУКИ І ОСВІТИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра обчислювальної техніки

Лабораторна робота №1

з Комп’ютерних систем

Виконав студенти групи ІО-11

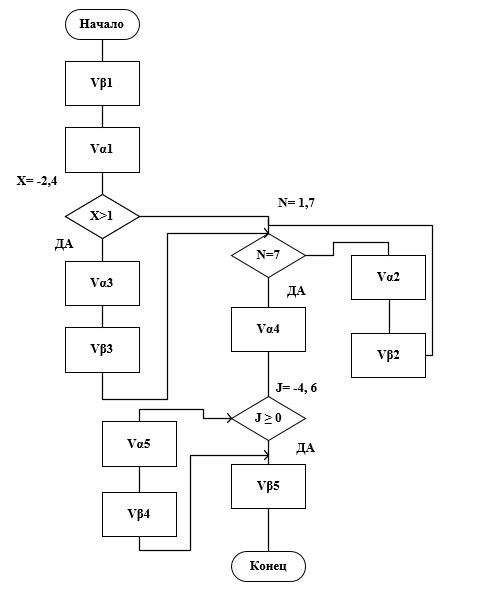
Ротенберг О.В

Бабак С.В

Номер групи - 3

Київ — 2014

Исходные данные



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | V9 | V10 | V11 | V12 |
| v1 |  |  |  |  |  |  |  |  |  |  |  |  |
| v2 |  |  | 1 |  |  |  |  |  |  |  |  |  |
| v3 |  |  |  | 3/7 |  | 24/49 |  | 4/49 |  |  |  |  |
| v4 |  |  |  |  | 1 |  |  |  |  |  |  |  |
| v5 |  |  |  |  |  | 6/7 |  | 1/7 |  |  |  |  |
| v6 |  |  |  |  |  |  | 1 |  |  |  |  |  |
| v7 |  |  |  |  |  | 6/7 |  | 1/7 |  |  |  |  |
| V8 |  |  |  |  |  |  |  |  | 5/11 |  | 6/11 |  |
| V9 |  |  |  |  |  |  |  |  |  | 1 |  |  |
| V10 |  |  |  |  |  |  |  |  |  |  | 1 |  |
| V11 |  |  |  |  |  |  |  |  |  |  |  |  |

Составим таблицу , определяющую для каждого оператора:

* число операций, составляющих оператор (K)
* среднее количество информации, передаваемое во время выполнения оператора (L)
* номер файла, с которым работает оператор (Nf)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | v1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 |
| *ki* |  | 0 | 56 | 110 | 0 | 84 | 0 | 140 | 26 | 0 | 0 |  |
| *li* |  | 90 | 0 | 0 | 100 | 0 | 139 | 0 | 0 | 67 | 140 |  |
| *h* |  | 1 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 2 | 3 |  |

Результаты программы :

Корни уравнения:

n[0] = 1.0

n[1] = 1.0

n[2] = 0.4287

n[3] = 0.4287

n[4] = 6.487600629811055

n[5] = 6.569200629811056

n[6] = 0.9999999999999998

n[7] = 0.4545999999999999

n[8] = 0.4545999999999999

n[9] = 0.9999999999999998

Среднее число процессорных операций, выполняемых при одном прогоне алгоритма: 799.9350529041287

Среднее число обращений к файлам:

Файл 1 : 7.569200629811056

Файл 2 : 0.8833

Файл 3 : 0.9999999999999998

Среднее количество информации передаваемой при одном обращении к файлам:

Файл 1 : 132.5263969874157

Файл 2 : 83.01618929016189

Файл 3 : 140.0

Сумма среднего числа обращений к основным операторам: 9.370900629811054

Средняя трудоемкость этапа: 85.36373231398335

Листинг программы :

**package** lab1;

**import** Jama.Matrix;

**public** **class** Lab01 {

**static** **int** *numberOfFiles* = 3;

**static** **double** [][] *left* = {{0,1,0,0,0,0,0,0,0,0},

{0,0,0.4287,0,0.4897,0.0816,0,0,0,0},

{0,0,0,1,0,0,0,0,0,0},

{0,0,0,0,0.8571,0,0.1429,0,0,0},

{0,0,0,0,0,1,0,0,0,0},

{0,0,0,0,0.8571,0,0.1429,0,0,0},

{0,0,0,0,0,0,0,0.4546,0,0.5454},

{0,0,0,0,0,0,0,0,1,0},

{0,0,0,0,0,0,0,0,0,1},

{0,0,0,0,0,0,0,0,0,0}};

**static** **double**[] *right* ={-1,0,0,0,0,0,0,0,0,0};

**static** **double**[] *answer* = **new** **double**[10];

**static** **double**[][] *somedata* = {{ 0,56,110,0,84,0,140,26,0,0}, //число операций, составляющих оператор

{90,0, 0,100, 0,139,0,0,67,140}, //среднее количество информации, передаваемое во время выполнения оператора

{1, 0, 0, 2, 0, 1, 0, 0, 2, 3}}; //номер файла, с которым работает оператор, если ноль = не работает с файлом

// Подсчет среднего числа процессорных операций, выполняемых при одном прогоне алгоритма

**public** **static** **double** processOperations(**double**[] answer1, **double** [][]somedata1){

**double** result = 0;

**for** (**int** i=0; i< answer1.length; i++){

result = result + answer1[i]\*somedata1[0][i];

}

**return** result;

}

// Подсчет среднего числа обращений к файлам

**public** **static** **double** fileRequest(**double**[] answer1, **double** [][]somedata1, **int** fileNumber){

**double** result = 0;

**for** (**int** i = 0; i< answer1.length; i++){

**if** (somedata1[2][i] == fileNumber)

result = result + answer1[i];

}

**return** result;

}

// Подсчет среднего количества информации передаваемой при одном обращении к файлам

**public** **static** **double** fileInformation(**double**[] answer1, **double** [][]somedata1, **int** fileNumber){

**double** result = 0;

**double** information =0;

**for** (**int** i = 0; i< answer1.length; i++){

**if** (somedata1[2][i] == fileNumber) {

result = result + answer1[i];

information = information + *somedata*[1][i]\*answer1[i];

}

}

information = information/result;

**return** information;

}

// Подсчет суммы среднего числа обращений к основным операторам

**public** **static** **double** operatorRequest(**double**[] answer1, **double** [][]somedata1){

**double** result = 0;

**for** (**int** i = 0; i< answer1.length; i++){

**if** (somedata1[0][i] !=0) result = result + answer1[i];

}

**return** result;

}

// Подсчет средней трудоемкости этапа

**public** **static** **double** middle(**double**[] answer1, **double** [][]somedata1){

**double** result = *processOperations*(answer1,somedata1);

**return** result/*operatorRequest*(answer1, somedata1);

}

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

Matrix A = **new** Matrix(*left*);

**for** (**int** j = 0; j < 10; j++) {

**for** (**int** q = j+1; q < 10; q++) {

**double** temp = A.get(j, q);

A.set(j, q, A.get(q, j));

A.set(q, j, temp);

}

}

**for** (**int** i =0; i< 10; i++)

A.set(i, i, A.get(i, i) -1);

Matrix B = **new** Matrix(*right*, 10);

Matrix result = A.solve(B);

**for** (**int** i =0; i< 10; i++)

*answer*[i] = result.get(i, 0);

System.***out***.println("Корни уравнения:");

**for** (**int** i=0;i<10;i++)

System.***out***.println(" n[" + i + "] = "+*answer*[i]);

System.***out***.println("Среднее число процессорных операций, выполняемых при одном прогоне алгоритма: " + *processOperations*(*answer*, *somedata*));

System.***out***.println("Среднее число обращений к файлам:");

**for** (**int** i = 1; i< *numberOfFiles*+1; i++)

{

System.***out***.println(" Файл " + i + " : "+ *fileRequest*(*answer*, *somedata*, i));

}

System.***out***.println("Среднее количество информации передаваемой при одном обращении к файлам:");

**for** (**int** i = 1; i< *numberOfFiles*+1; i++)

{

System.***out***.println(" Файл " + i + " : "+ *fileInformation*(*answer*, *somedata*, i));

}

System.***out***.println("Сумма среднего числа обращений к основным операторам: " + *operatorRequest*(*answer*, *somedata*));

System.***out***.println("Средняя трудоемкость этапа: " + *middle*(*answer*, *somedata*));

}

}