**Національний технічний університет Україні «Київський політехнічний інститут»**

**Кафедра обчислювальної техніки**

***Лабораторна робота №6***

***з курсу «Теорія імовірності та елементи математичної статистики»***

***Виконав:***

***Студент групи ІО-02***

***Малижев Богдан Сергійович***

**Київ 2011**

***Варіант* *завдання***

***Неперервний Марківський процес з 5-ма станами***

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** org.apache.commons.math.linear.Array2DRowRealMatrix;

**import** org.apache.commons.math.linear.RealMatrix;

**public** **class** Main {

**private** **static** **void** simulate(**double**[][] switchTable, **int** vertex, **int** cycleCount) {

**double**[] vertexProbability = **new** **double**[switchTable.length];

System.*out*.println("Починаючи з вершини "+(vertex+1));

**for**(**int** i = 0; i<cycleCount; i++){

vertex = *decide*(switchTable[vertex], vertex, vertexProbability);

//System.out.println("Switched to "+(vertex+1));

}

**double** S = 0;

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

S += vertexProbability[i];

//System.out.println(vertexProbability[i]);

}

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

System.*out*.println("Система знаходилась в "+(i+1)+" стані "+(vertexProbability[i]/S)+" часу");

}

}

**private** **static** **int** decide(**double**[] switchTable, **int** currentVertex, **double**[] timeTable){

**double**[] time = **new** **double**[switchTable.length];

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

time[i] = -Math.*log*(Math.*random*())/switchTable[i];

//System.out.println(time[i]);

}

**int** result = *findMin*(time);

timeTable[currentVertex] += time[result];

**return** result;

}

**private** **static** **int** findMin(**double**[] time){

**int** result = 0;

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

**if**(time[i]<time[result]){

result = i;

}

}

**return** result;

}

**private** **static** **void** printStationaryP(**double**[][] switchTable) {

/\*double p;

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

p = 0;

for(int j = 0; j<switchTable.length; j++){

p+= switchTable[j][i];

}

System.out.println("P"+(i+1)+" = "+p/switchTable.length);

}\*/

**double**[][] array1 = **new** **double**[switchTable.length][switchTable.length];

**double**[][] array2 = **new** **double**[switchTable.length][1];

array2[0][0] = 1;

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

array1[0][i] = 1;

}

**for**(**int** i=1; i<switchTable.length; i++){

**for**(**int** j=0; j<switchTable.length; j++){

**if**(i == j){

**for**(**int** k = 0; k<switchTable.length; k++){

**if**(k != j){

array1[i][j] -= switchTable[j][k];

}

}

}**else**{

array1[i][j] = switchTable[j][i];

}

}

}

//System.out.println(Arrays.deepToString(switchTable));

//System.out.println(Arrays.deepToString(array1));

//System.out.println(Arrays.deepToString(array2));

RealMatrix mat1 = **new** Array2DRowRealMatrix(array1);

RealMatrix mat2 = **new** Array2DRowRealMatrix(array2);

mat1 = mat1.~~inverse~~();

**double**[][] result = mat1.multiply(mat2).getData();

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

System.*out*.println("P"+(i+1)+" = "+result[i][0]);

}

}

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

**double**[][] switchTable = **null**;

**int** dim;

**int** startVertex = 0;

**int** cycleCount = 0;

**try**{

BufferedReader stream = **new** BufferedReader(**new** InputStreamReader(System.*in*));

System.*out*.println("Кількість вершин:");

dim = Integer.*parseInt*(stream.readLine());

switchTable = **new** **double**[dim][dim];

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

**for**(**int** j = 0; j<switchTable.length; j++){

**if**(i != j){

System.*out*.println("Лямбда від "+(i+1)+" до "+(j+1));

switchTable[i][j] = Double.*parseDouble*(stream.readLine());

}

}

}

System.*out*.println("Початкова вершина:");

startVertex = Integer.*parseInt*(stream.readLine())-1;

//System.out.println("Enter cycles count");

//cycleCount = Integer.parseInt(stream.readLine());

cycleCount = 100000;

}**catch**(Exception e){

e.printStackTrace();

}

*printStationaryP*(switchTable);

*simulate*(switchTable, startVertex, cycleCount);

}

}