Лабораторная работа №2

# Графы

***Составить программы, решающие следующие задачи. Входные данные находятся в файле INPUT.TXT. Для каждой задачи изобразить на экране соответствующий граф.***

19 вариант.

В компьютерной игре герою требуется пройти через все залы, уничтожив в каждом по одному монстру. Монстры имеют различную силу, и убить каждого из них можно, если сила монстра не превосходит силы героя. Сила героя каждый раз возрастает на величину силы убитого монстра. Известны, сила каждого монстра, начальная сила героя, а также между какими залами имеются двери (через дверь можно пройти только в одном направлении). Определить все пути, двигаясь по которым, герой сможет убить всех монстров, побывав в каждом зале только один раз.

Технические требования:

Количество залов N≤10. Залы пронумерованы числами от 1 до N.

Формат файла INPUT.TXT:

1-я строка – количество залов; 2-я - начальная сила героя; 3-я – N натуральных чисел через пробел – силы монстров в каждом зале; последующие строки содержат по два натуральных числа – номера залов X и Y, между которыми существует дверь из X в Y.

* Создание функции обхода в глубину с параметрами (массив матрицы смежности a, массив меток metka, массив вывода out, индекс записи в массив вывода index, номер иерархии пути num, номер строки матрицы смежности i, количество залов n, сила героя power, массив силы монстров monsters\_power):
* metka[i] = ++num, out[++index] = i + 1
* Создание двух массивов-буферов, копирование в первый массива меток и во второй массива вывода для сохранения их текущих значений
* Для j = 0, j < n, ++j
  + - Если существует путь из комнаты i в комнату j и metka[j] == 0 и сила героя больше или равна силе монстра в комнате:
      * Вызов функции обхода (a, metka, out, index, num, j, n, power + monsters\_power[j], monsters\_power)
      * Копирование массивов-буферов в массивы матрицы смежности и вывода обратно
* Вывод пути, если index == n – 1 (массива вывода заполнен)
* Создание матрицы смежности “a”: двумерного массива размером n2, где n – количество залов, для хранения наличия пути из одного зала в другой
* Открытие текстового файла; построчное чтение количества залов “n”, начальной силы героя “power”, силы монстров в каждом зале monsters\_power, путей между залами в массив “a”; закрытие файла
* Создание массива metka размером в количество залов для хранения информации о посещённых комнатах, переменной num=0 для определения порядка посещения комнаты
* Для i = 0, i<n, i++:
  + Обнуление массива меток
  + Создание массива out размером в количество залов для хранения выходного пути, создание переменной index=-1 для определения места записи в массив out
  + Вызов функции обхода (a, metka, out, index, num, i, n, power + monsters\_power[i], monsters\_power)

void obhod(int a[][100], int\* metka, int\* out, int index, int num, int i, int n, int power, int \* monsters\_power) {

metka[i] = ++num;

out[++index] = i + 1;

//буффер меток

int\* metka2{ new int[n] };

//буффер вывода

int\* out2{ new int[n] };

for (int r = 0; r < n; ++r)

metka2[r] = metka[r], out2[r] = out[r];

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

if (a[i][j] == 1 && metka[j] == 0 && power >= monsters\_power[j]) {

power += monsters\_power[j];

obhod(a, metka, out, index, num, j, n, power, monsters\_power);

for (int r = 0; r < n; ++r)

metka[r] = metka2[r], out[r] = out2[r];

power -= monsters\_power[j];

}

delete[] metka2;

if (index == n - 1) {

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

if (j) textBox5->Text += "->";

textBox5->Text += out[j];

}

textBox5->Text += Environment::NewLine;

}

}

private: System::Void button\_start\_Click(System::Object^ sender, System::EventArgs^ e) {

StreamReader^ f = gcnew StreamReader("input.txt");

textBox5->Text = "";

textBox4->Text = "";

//массив наличия пути

int a[100][100] = { 0 };

//число залов

textBox1->Text = f->ReadLine();

int n = Convert::ToInt32(textBox1->Text);

//стартовая сила

textBox2->Text = f->ReadLine();

int power = Convert::ToInt32(textBox2->Text);

//сила монстров через пробел

textBox3->Text = f->ReadLine();

array <String^>^ monstrs = textBox3->Text->Split(' ');

int\* monsters\_power{ new int[n] };

//односторонние двери построчно

String^ buffer = f->ReadToEnd();

array <String^>^ doors = buffer->Split('\n', ' ');

f->Close();

//запись в массив наличия путей между дверьми и силы монстров

for (int i = 0; i < doors->Length; i += 2) {

textBox4->Text += doors[i] + " " + doors[i + 1] + Environment::NewLine;

a[Convert::ToInt32(doors[i]) - 1][Convert::ToInt32(doors[i + 1]) - 1] = 1;

}

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

monsters\_power[i] = Convert::ToInt32(monstrs[i]);

}

//вывод наличия путей

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

textBox5->Text += (i + 1) + "\t";

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

textBox5->Text += a[i][j] + " ";

textBox5->Text += Environment::NewLine;

}

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

textBox5->Text += monsters\_power[j] + " ";

textBox5->Text += Environment::NewLine;

//массив посещённых узлов

int\* metka{ new int[n] };

//номер посещённого узла

int num = 0;

//запуск обхода

textBox5->Text += Environment::NewLine;

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

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

metka[j] = 0;

//массив вывода

int\* out{ new int[n] };

int index = -1;

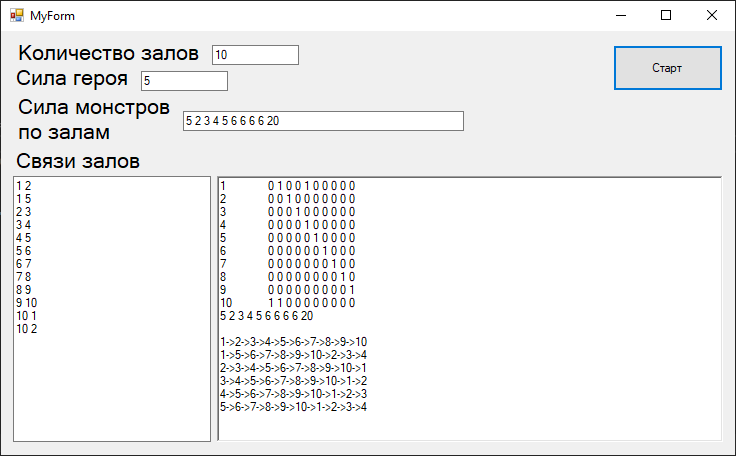
if (power >= monsters\_power[i])

obhod(a, metka, out, index, num, i, n, power + monsters\_power[i], monsters\_power);

delete[] out;

}

}



Работа не моей программы

