

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительных систем

ОТЧЁТ

по лабораторной работе №5
по дисциплине «Моделирование»
на тему «Генерация случайных деревьев»

Выполнил:
студент группы ИВ-222
Терешков Р. В.

Проверил:
д.т.н., доцент
Родионов А. С.

Новосибирск – 2016

Задание

В рамках данной лабораторной работы необходимо реализовать алгоритм генерации случайных деревьев. В соответствии с вариантом дерево строится с ограничением на максимальную степень вершины при помощи метода допустимого выбора (МДВ).

Ход работы и результаты моделирования

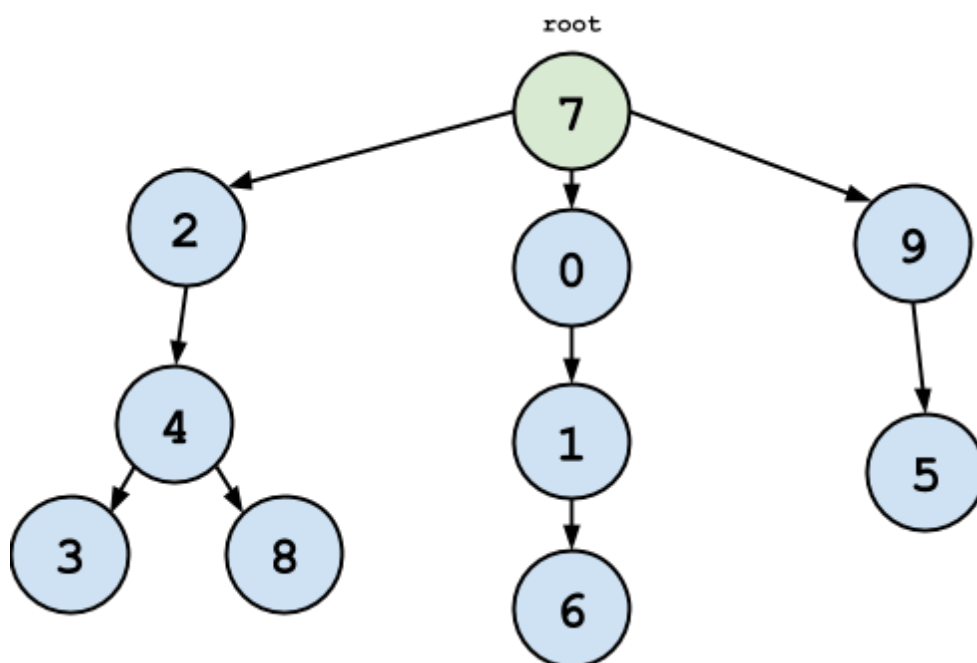
При построении дерева используются следующие параметры:

- N – количество вершин в дереве;
- DEG – максимальная степень вершины;
- $root$ – индекс корня дерева;
- $Degree[0..N - 1]$ – массив степеней каждой из вершин;
- $io[0..N - 2]$ – массив индексов конечных вершин для рёбер;
- $ou[0..N - 2]$ – массив индексов начальных вершин для ребёр.

Результатом моделирования является случайно сгенерированное дерево с максимальной степенью вершины $Degree_i \leq DEG$.

i	0	1	2	3	4	5	6	7	8	9
Degree _i	2	2	2	1	3	1	1	3	1	2

ou	7	7	7	2	9	0	1	4	4	
io	9	2	0	4	5	1	6	3	8	



Листинг программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10
#define DEG 4

int drand(int from, int to)
{
    return from + rand() % (to - from + 1);
}

void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

int main()
{
    printf("\nMax degree: %d\nNumber of vertices: %d\n\n", DEG, N);
    srand((unsigned)time(NULL));

    int k = 0;
    int I[N], Degree[N];
    int io[N - 1], ou[N - 1];

    for (int i = 0; i < N; ++i) {
        I[i] = i;
        Degree[i] = 0;
    }

    int root = rand() % (N - 1);

    printf("Root: %d\n\n", root);

    swap(&I[N - 1], &I[root]);

    for (int i = 0; i < N - 1; ++i) {
        int s = drand(N - 1 - i, N - 1 - k);
        int t = drand(0, N - 2 - i);
```

```

    io[i] = I[t];
    ou[i] = I[s];

    ++Degree[I[t]];
    ++Degree[I[s]];

    if (Degree[I[s]] == DEG) {
        I[s] = I[N - 1 - k];
        ++k;
    }
    swap(&I[N - 2 - i], &I[t]);
}

printf("DEGREE\n");
for (int i = 0; i < N; ++i) printf("[%d]: %d ", i, Degree[i]);
printf("\nOU\n");
for (int i = 0; i < N - 1; ++i) printf("[%d]: %d ", i, ou[i]);
printf("\nIO\n");
for (int i = 0; i < N - 1; ++i) printf("[%d]: %d ", i, io[i]);
printf("\n");

return 0;
}

```