Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №9**

По дисциплине «Математические основы интеллектуальных систем»

Тема: «Выборки (перестановки, сочетания, размещения)»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Козинский А. А.

Брест 2023

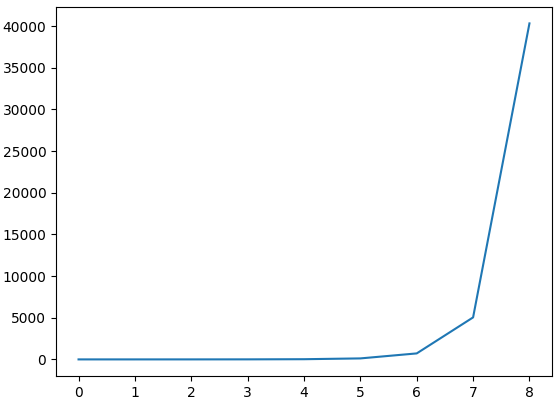
**Цель:** познакомиться с алгоритмом перебора высокой сложности.

**Ход работы:**

**Вариант 7**

1. Реализовать алгоритм построения ВСЕХ перестановок N натуральных чисел. Выполнить все пункты указаний к тестированию. Оценить максимальное N, для которого ваш личный ноутбук выполняет такую реализацию за время меньшее (120 секунд ).
2. Подсчитать количество натуральных чисел в интервале от 1 до n, которые делятся на a, b или с. Значения n, a, b, с задаются входными данными (файл).
3. Сколько существует перестановок из n элементов, в которых хотя бы один элемент находится на своем месте. Перестановки вывести.

**Задание 1**



#include <iostream>

#include <time.h>

using namespace std;

bool NextSet(int \*a, int n){

int j = n - 2;

while (j != -1 && a[j] >= a[j + 1])

j--;

if (j == -1)

return false; // больше перестановок нет

int k = n - 1;

while (a[j] >= a[k])

k--;

swap(a[j], a[k]);

int l = j + 1, r = n - 1; // сортируем оставшуюся часть последовательности

while (l < r)

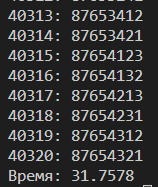
swap(a[l++], a[r--]);

return true;

}

int main(){ unsigned short N; unsigned iter = 0;

cout << "Введите N: "; cin >> N;



int arr[N];

for (short i = 0; i < N; i++){

arr[i] = i + 1;

}

struct timespec start, end;

clock\_gettime(CLOCK\_REALTIME, &start);

do{

iter++;

cout << iter << ": ";

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

cout << arr[x];

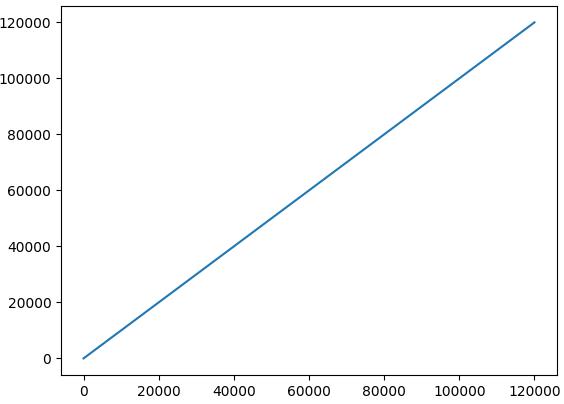
cout << endl;

} while (NextSet(arr, N));

clock\_gettime(CLOCK\_REALTIME, &end);

cout << "Время: " << (end.tv\_sec - start.tv\_sec) + (end.tv\_nsec - start.tv\_nsec) / 1000000000.0;

}



**Задание 2**

#include <iostream>

#include <fstream>

#include "../get\_time.h"

using namespace std;

int main(){

ifstream file("E:\\Studing\\MOIS\\Lab. 9\\task2");

long long n, a, b, c;

unsigned div\_a = 0, div\_b = 0, div\_c = 0;

file >> n >> a >> b >> c;

start\_clock();

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

if (i % a == 0)

div\_a++;

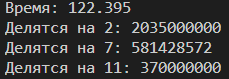
if(i % b == 0)

div\_b++;

if(i % c == 0)

div\_c++;

}

 stop\_clock(); cout << "Делятся на " << a << ": " << div\_a << endl;

cout << "Делятся на " << b << ": " << div\_b << endl;

cout << "Делятся на " << c << ": " << div\_c << endl;

}

**Задание 3**

#include <iostream>

#include "../get\_time.h"

using namespace std;

bool NextSet(int \*a, int n){

    int j = n - 2;

    while (j != -1 && a[j] >= a[j + 1])

        j--;

    if (j == -1)

        return false; // больше перестановок нет

    int k = n - 1;

    while (a[j] >= a[k])

        k--;

    swap(a[j], a[k]);

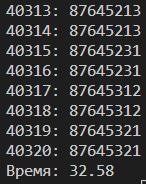
    int l = j + 1, r = n - 1; // сортируем оставшуюся часть последовательности

    while (l < r)

        swap(a[l++], a[r--]);

    return true;

}

int main(){

    unsigned short N; unsigned iter = 0;

    cout << "Введите N: "; cin >> N;

    int arr[N];

    for (short i = 0; i < N; i++){

        arr[i] = i + 1;

    }

    start\_clock();

    do{

        for (int x = 0; x < N; x++)

            if(arr[x] == x + 1){

                cout << ++iter << ": ";

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

                    cout << arr[z];

                cout << endl;

            }

    } while (NextSet(arr, N));

    stop\_clock();

}

**Вывод:** в ходе лабораторной работы я научился находить кратчайшие пути в графе.