Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №9

По дисциплине МОИС за 2 семестр

«ВЫБОРКИ (ПЕРЕСТАНОВКИ, СОЧЕТАНИЯ, РАЗМЕЩЕНИЯ)»

Выполнил

Студент группы ИИ-23

Романюк А.П.

Проверил

Козинский А.А.

Брест 2024

Задача 1, 3

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).

Сколько существует перестановок из n элементов, в которых только хотя бы два элемента находятся на своих местах. Перестановки вывести.

#include <iostream>

#include <vector>

#include <ctime>

using namespace std;

bool gen(vector<int>& a, int n) {

int j = a.size() - 2;

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

if (j == -1) return false;

int k = n - 1;

while (a[k] < a[j]) k--;

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

int l = j + 1;

int r = n - 1;

while (l < r) {

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

l++;

r--;

}

return true;

}

void print(vector<int> a) {

//static int num = 1;

cout.width(3);

//cout << num++ << ": ";

for (int i = 0; i < a.size(); i++)

cout << a[i] << " ";

cout << endl;

}

void hasTwoFixed(vector<int> a) {

for (int i = 0; i < a.size(); i++) {

if (a[i] == i + 1) {

for (int j = i + 1; j < a.size(); j++) {

if (a[j] == j + 1) {

print(a);

return;

}

}

}

}

}

int main() {

vector<int> a;

int time = 0;

for (int n = 4; time < 1; n++) {

unsigned int start = clock();

for (int i = 1; i <= n; i++) {

a.push\_back(i);

}

//print(a, n);

hasTwoFixed(a);

while (gen(a, n)) {

//print(a, n);

hasTwoFixed(a);

}

unsigned int end = clock();

time = (end - start) / 1000;

cout << endl << "n = " << n << " time = " << time << "s.\n";

a.clear();

}

}

Задача 2

Найти первую пару чисел n и k (n + k имеет минимальное значение), при которых превысит значение 10m (n, k, m – задаются как входные данные).

#include <iostream>

#include <cmath>

using namespace std;

int factorial(int n) {

int f = 1;

for (int i = 1; i <= n; i++) {

f \*= i;

}

return f;

}

int calculateAnk(int n, int k) {

if (factorial(n - k) > 0)

return factorial(n) / factorial(n - k); else return -1;

}

pair<int, int> findMinPair(int m) {

int minSum = INT\_MAX;

std::pair<int, int> minPair = { 0, 0 };

for (int n = 1; ; ++n) {

for (int k = 1; k <= n; ++k) {

int currentSum = calculateAnk(n, k);

if (currentSum > pow(10, m) && currentSum < minSum) {

minSum = currentSum;

minPair = { n, k };

return minPair;

} else if (calculateAnk(n, k) == -1) return minPair = { 0, 0 };

}

}

}

int main() {

int mValue = 2; // Задайте нужное значение m

std::pair<int, int> result = findMinPair(mValue);

cout <<"for m = " << mValue << " first pair n and k: (" << result.first << ", " << result.second << ")\n";

return 0;

}