**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ**

**Институт компьютерных технологий и информационной безопасности**

**Кафедра математического обеспечения и применения ЭВМ**

**Отчет по лабораторной работе №5**

**по курсу «Структуры и алгоритмы обработки данных»**

**на тему**

**«Решение комбинаторных задач методом перебора»**

**Вариант №8**

Выполнил: студент

группы КТбо2-1: Самардак А.В

Проверил:

Доцент каф. МОП ЭВМ Дроздов С.Н

**Цель работы**

Знакомство с общей постановкой комбинаторных задач, освоение техники решения таких задач методом перебора и приемов анализа решения.

**Формулировка задачи**

Во всех вариантах задания требуется написать и отладить программу на произвольном языке программирования, решающую методом перебора комбинаторную задачу, выбранную в соответствии с вариантом задания. Желательно учесть все соображения, позволяющие сократить перебор.

*Задача о трех станках*:

Даны N деталей, каждая из которых должна быть обработана на станке A, затем на станке B, затем на станке C. Каждый станок может в данный момент времени обрабатывать только одну деталь. Если нужный станок занят, то другие детали могут ожидать его освобождения. Для каждой детали известны длительности ее обработки на каждом станке: TA i , TB i , TC i . Требуется найти такой порядок запуска деталей на обработку (т.е. найти такую перестановку номеров деталей), при котором длительность обработки всего комплекта деталей минимальна.

**Листинг программы на языке С++**

#include <iostream>

#include <vector>

#include <queue>

#include <ctime>

#include <algorithm>

struct Detail

{

int ta;

int tb;

int tc;

};

std::vector<Detail> vec;

int64\_t cnt = 0;

int detail\_procesing(std::vector<int>& order)

{

cnt++;

int res = 0;

std::queue<Detail> a;

std::queue<Detail> b;

std::queue<Detail> c;

for (int i : order)

{

a.push(vec[i]);

}

while (!(a.empty() && b.empty() && c.empty()))

{

int tmp = INT32\_MAX;

tmp = (a.empty() ? tmp : std::min(tmp, a.front().ta));

tmp = (b.empty() ? tmp : std::min(tmp, b.front().tb));

tmp = (c.empty() ? tmp : std::min(tmp, c.front().tc));

if (!a.empty())

a.front().ta -= tmp;

if (!b.empty())

b.front().tb -= tmp;

if (!c.empty())

c.front().tc -= tmp;

if (!a.empty() && a.front().ta == 0)

{

b.push(a.front());

a.pop();

}

if (!b.empty() && b.front().tb == 0)

{

c.push(b.front());

b.pop();

}

if (!c.empty() && c.front().tc == 0)

{

c.pop();

}

res += tmp;

}

return res;

}

std::vector<Detail> randomize(int n, int d)

{

std::vector<Detail> res(n);

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

{

res[i].ta = 1 + (rand() % d);

res[i].tb = 1 + (rand() % d);

res[i].tc = 1 + (rand() % d);

}

return res;

}

void print(int result\_time, int ans, std::vector<int> arr)

{

std::cout << "Время работы: " << result\_time << "мс" << std::endl;

std::cout << "Колличество вызовов: " << cnt << std::endl;

std::cout << ans << std::endl;

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

{

std::cout << arr[i] + 1 << " | ";

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int n;

std::cin >> n;

vec = std::vector<Detail>(10);

std::vector<int> res(n);

int d;

std::cin >> d;

vec = randomize(n, d);

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

{

res[i] = i;

}

std::vector<int> tmp = res;

unsigned int start = clock();

int ans = INT32\_MAX;

do

{

int count = detail\_procesing(tmp);

if (count < ans)

{

res = tmp;

ans = count;

}

} while (std::next\_permutation(tmp.begin(), tmp.end()));

unsigned int stop = clock();

print(stop - start, ans, res);

return 0;

}

**Заключение**

В ходе выполнения данной лабораторной работы была решена задача о трёх станках при помощи метода перебора. Ниже представлен прогон программы на пяти случайных тестах для каждого N, с вычислением количества вызовов, а также минимального, максимально и среднего «минимального пути» (в моём случае это у.е. времени по условию задачи) для решения этой задачи.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Среднее время, мс | Количество вызовов | Минимальный путь | | |
| минимальное | среднее | максимальное |
| 3 | 0,0 | 6 | 242 | 270 | 360 |
| 4 | 2,4 | 24 | 270 | 325 | 394 |
| 5 | 17,6 | 120 | 344 | 395 | 450 |
| 6 | 126 | 720 | 442 | 464 | 493 |
| 7 | 1119 | 5040 | 487 | 508 | 555 |
| 8 | 10362 | 40320 | 487 | 545 | 612 |
| 9 | 108373 | 362880 | 553 | 594 | 673 |