Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного

обеспечения и администрирования

информационных систем

Направление подготовки

математическое обеспечение и

администрирование

информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №11**

«Решение комбинаторных задач методом перебора»

Выполнил:

студент группы 213 Тихонов Е.Е.

Проверил:

профессор кафедры ПОиАИС Кудинов В.А.

Курск, 2022

**Цель работы**: знакомство с общей постановкой комбинаторных задач, освоение техники решения таких задач методом перебора и приемов анализа решения.

**Задание**

Задача о куче камней. Имеется N камней, известны их веса Pi (i=1...N), задано количество куч M. Требуется разложить камни на M куч так, чтобы минимизировать вес самой тяжелой кучи. (Вариант формулировки: даны N программ с длительностями Pi и M процессоров. Требуется распределить программы так, чтобы раньше закончить выполнение).

Время работы программы для разных наборов данных представлены в таблице 1.

Таблица 1 - Время работы программы для разных наборов данных

|  |  |  |  |
| --- | --- | --- | --- |
| **N** | **M** | **Среднее время** | **Количество итераций** |
| 4 | 2 | 30 мс | 71 |
| 5 | 2 | 55 мс | 175 |
| 6 | 2 | 110 мс | 415 |
| 4 | 3 | 80 мс | 333 |
| 5 | 3 | 250 мс | 1255 |
| 6 | 3 | 790 мс | 4495 |
| 5 | 4 | 800 мс | 5205 |

**Код программы**

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

int IterationCount = 0;

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Введите количество камней N = ";

cin >> n;

int\* weight = new int[n];

int\* id = new int[n];

cout << "Введите веса соответствующие камням\n";

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

{

cin >> weight[i];

id[i] = 0;

}

cout << "\nВведите количество куч M = ";

cin >> m;

int MinSum = 0; //минимальный вес самой тяжелой кучи

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

{

MinSum += weight[i];//изначально присвоим значение суммы весов всех камней

}

cout << "\nВсевозможные комбинации размещения камней в кучах\nпервый столбец: позиция соответствует камню, а число куче\nвторой столбец: веса квмней распределенные по кучам\n";

while (id[0] == 0)

{

//выводим комбинацию

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

{

cout << id[i] << " ";

}

int MaxSum = 0; //вес самой тяжелой кучи

for (int mi = 0; mi < m; mi++)

{

cout << " (";

int sum = 0;

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

{

IterationCount++;

if (id[i] == mi)

{

sum += weight[i];

cout<< weight[i] << " ";

}

}

if (sum == 0)

cout << ")";

else

cout << "\b)";

if (sum > MaxSum)

{

MaxSum = sum;

}

}

cout << " наибольший вес кучи = " << MaxSum << endl;

if (MinSum > MaxSum)

{

MinSum = MaxSum;

}

id[n - 1]++;

int i = n - 1;

while (id[i] >= m)

{

IterationCount++;

int temp = id[i];

id[i] = temp % m;

id[i - 1] += temp / m;

i--;

}

}

cout << "минимальный вес самой тяжелой кучи = " << MinSum << endl;

cout << "Количество итераций " << IterationCount << endl;

return 0;

}

**Тест программы**

Тест 1 программы для решения задачи представлен на рисунке 1. Тест 2 программы для решения задачи представлен на рисунке 2.

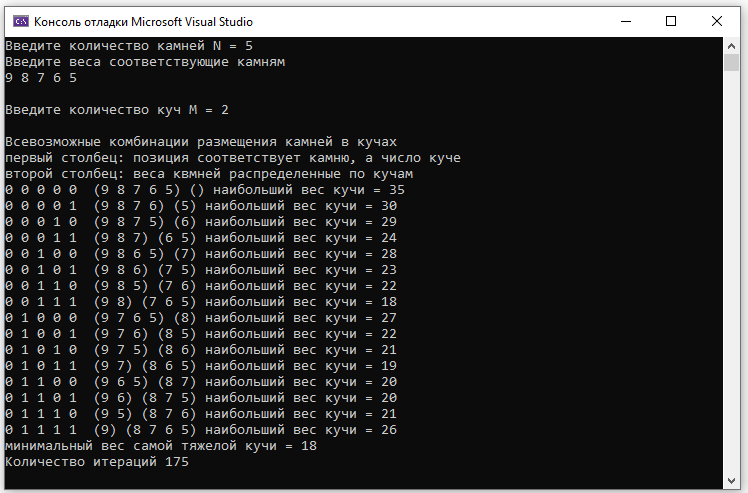


Рисунок 1 – Тест 1 программы для решения задачи

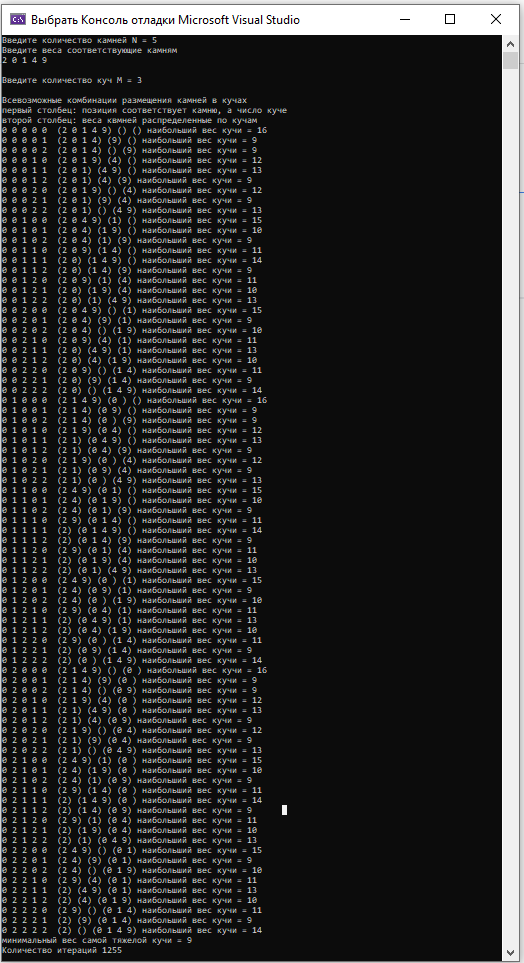


Рисунок 2 – Тест 2 программы для решения задачи