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 №12**

«Решение комбинаторных задач с помощью эвристических алгоритмов»

Выполнил:

студент группы 213 Тихонов Е.Е.

Проверил:

профессор кафедры ПОиАИС Кудинов В.А.

Курск, 2022

**Цель работы**: изучение эвристических алгоритмов и способов их разработки, а также сравнительная оценка использования эвристических и переборных алгоритмов для решения комбинаторных задач.

**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 и сравнение эвристического алгоритма с алгоритмом перебора представлены в таблице 1.

Таблица 1 - Время работы программы для разных наборов данных. Сравнение эвристического и переборного методов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N** | **M** | **Среднее время** | | **Количество итераций** | |
| **Перебор** | **Эвристика** | **Перебор** | **Эвристика** |
| 4 | 2 | 30 мс | 3 мс | 71 | 8 |
| 5 | 2 | 55 мс | 5 мс | 175 | 11 |
| 6 | 2 | 110 мс | 7 мс | 415 | 14 |
| 4 | 3 | 80 мс | 3 мс | 333 | 7 |
| 5 | 3 | 250 мс | 5 мс | 1255 | 11 |
| 6 | 3 | 790 мс | 7 мс | 4495 | 15 |
| 5 | 4 | 800 мс | 4 мс | 5205 | 9 |

**Код программы**

#include <iostream>

#include <cmath>

#include <stdlib.h>

using namespace std;

int IterationCount = 0;

int getMinStack(int\* stack\_weight, int stack\_count)

{

int min\_stack = 0;

for (int i = 0; i < stack\_count; i++)

{

IterationCount++;

if (stack\_weight[min\_stack] > stack\_weight[i])//находим минимальную кучу

{

min\_stack = i;

}

}

return min\_stack;

}

int main()

{

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Введите количество камней N = ";

cin >> n;

int\* stone\_weight = new int[n];

int\* stone\_id = new int[n];

cout << "Введите веса соответствующие камням\n"; //Веса должны быть расположены по убыванию либо нужно применить сортировку

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

{

cin >> stone\_weight[i];

stone\_id[i] = 0;

}

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

cin >> m;

int\* stack\_weight = new int[m];

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

{

stack\_weight[i] = 0;

}

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

{

IterationCount++;

int min\_stack = 0;

if (i < m)

{

min\_stack = i;

}

else if(i == m)

{

min\_stack = i-1;

}

else

{

min\_stack = getMinStack(stack\_weight, m);

}

stone\_id[i] = min\_stack;

stack\_weight[min\_stack] += stone\_weight[i];

}

int MaxSum = 0;

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

{

IterationCount++;

if (stack\_weight[i] > MaxSum)

{

MaxSum = stack\_weight[i];

}

}

cout << "минимальный вес самой тяжелой кучи = " << MaxSum << 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.

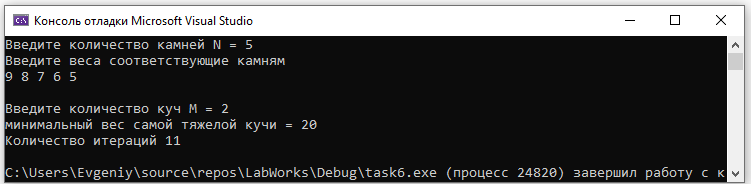


Рисунок 1 – Тест 1 программы для решения задачи

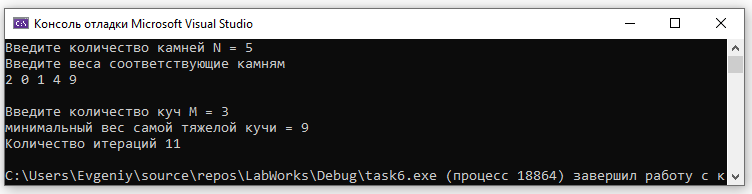


Рисунок 2 – Тест 2 программы для решения задачи