Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №2 (1005)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

# Цель работы

Решить задачу №1005 на платформе Timus Online Judge <https://acm.timus.ru/problem.aspx?space=1&num=1005>

# Задача

## Условие

У вас есть несколько камней известного веса w1, …, wn. Напишите программу, которая распределит камни в две кучи так, что разность весов этих двух куч будет минимальной.

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

## Исходные данные

Ввод содержит количество камней n (1 ≤ n ≤ 20) и веса камней w1, …, wn (1 ≤ wi ≤ 100 000) — целые, разделённые пробельными символами.

## Результат

Ваша программа должна вывести одно число — минимальную разность весов двух куч

## Пример

|  |  |
| --- | --- |
| **Исходные данные** | **Результат** |
| 5 5 8 13 27 14 | 3 |

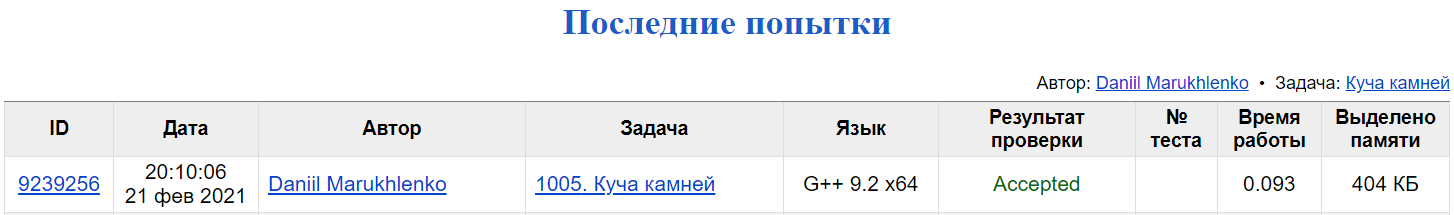
# Материалы работы

* 1. Объяснение алгоритма

По условию количество общее количество камней не превосходит 20, следовательно можно использовать перебор всех возможных комбинаций. Это можно сделать рекурсивной, например, функцией или битовой «маской» на массив. В реализованном алгоритме в цикле вводится цикл, значение переменной счётчика цикла идёт от 0 до 2^n-1, на каждой итерации рассматривается двоичное представление счётчика. Каждому биту сопоставляется элемент массива весов камней. В зависимости от значения бита решается камень будет лежать в первой куче или во второй. В процессе выполнения цикла сохраняется минимальная разность весов двух куч, а после его завершения число выводится в ответ.

* 1. Код программы.

1. *#include <iostream>*
2. *#include <cmath>*
3. *using namespace std;*
4. *int main(){*
6. *int n, sum = 0;*
7. *cin >> n;*
8. *int array[n];*
9. *for (int i = 0; i<n; i++){*
10. *cin>>array[i];*
11. *sum += array[i];*
12. *}*
13. *int answer = sum;*
14. *int max\_n\_mask = pow(2,n)-1;*
15. *int temp\_sum;*
16. *int delta;*
17. *for (int i = 0; i < max\_n\_mask; i++){*
18. *temp\_sum = 0;*
19. *for (int j = 0; j<n; j++){*
20. *if(i & (1 << j)) temp\_sum += array[j];*
21. *delta = abs(sum-temp\_sum\*2);*
22. *if (delta<answer) answer = delta;*
23. *}*
24. *}*
25. *cout << answer << endl;*
26. *return 0;*
27. *}*
28. Результат выполнения и ссылка на репозиторий GitHub



<https://github.com/japersik/algorithms_and_data_structures/tree/master/task_2_1005>

# Вывод

Работа выполнена, задача была решена перебором всех возможных вариантов.