

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Найти максимальную сумму значений вершин графа.»

Выполнил работу

Воробьев Егор

Академическая группа J3113

Принято

Ассистент, Дунаев Максим

Санкт-Петербург

2024

1. Введение

Цель: Найти максимальную сумму значений вершин графа.

Задачи:

- 1) Написать оптимизированный код с применением жадного алгоритма

2. Теоретическая подготовка

Типы данных:

- 1) `std::vector` – для реализации двух массивов текущего шага и предыдущего;

Заголовочные файлы:

- 1) `<vector>` - для использования типа данных `std::vector`;

3. Реализация

- 1) Для начала заметим, что применение операции меняет значение двух вершин. То есть после произвольного количества выполнения операции измененных вершин будет четное количество. В определенные моменты их количество может быть нечетным, это значит, что какая-то измененная вершина “лишняя”. Будем опираться на это при решении.
- 2) По условию нужно считать ребра, но из пункта 1) следует, что можно считать для каждой вершины отдельно. Чтобы учесть необходимость четного количества вершин, мы введем две переменные, которые отвечают за минимальную разницу между измененной вершиной и ее неизменной версии. Перебираем вершины, если измененное значение больше предыдущего, то увеличиваем счетчик измененных вершин, добавляем значение измененной вершины в итоговую сумму и вычисляем минимальную разницу. Если измененное значение меньше, то добавляем предыдущее значение в сумму и вычисляем минимальную разницу.

```

long long maximum_value_sum(vector<int>& nums, int k, vector<vector<int>>& edges) {
    int n = nums.size();
    long long ans = 0;
    int xor_d = 1e9;
    int nxor_d = 1e9;
    int cnt = 0;
    for (int i = 0 ; i < n ; i++) {
        int xor_val = nums[i]^k;
        if (xor_val > nums[i]) {
            cnt++;
            ans += xor_val;
            xor_d = min(xor_d , xor_val - nums[i]);
        } else {
            ans += nums[i];
            nxor_d = min(nxor_d , nums[i]-xor_val);
        }
    }
}

```

Изображение №1 – Перебор вершин

- 3) В конце проверяем количество измененных вершин на четность, если условие выполняется, то итоговая сумма является максимальной. Если условие не выполняется, то это значит, что одна из вершин лишняя. Поскольку нам нужна максимальная сумма, то вычитаем минимальную из разниц, как бы “откатывая” назад значение одной вершины.

```

if (cnt%2 == 0)
    return ans;

return ans - min(xor_d, nxor_d);

```

Изображение №2 – Проверка на четность

4. Заключение

В ходе выполнения работы мною было реализовано оптимизированное решение задачи с применением жадного алгоритма с платформы leetcode. Цель работы была достигнута путем тестов на сайте, которые были пройдены.

5. Приложения

ПРИЛОЖЕНИЕ 1

Основной код алгоритма.

```
long long maximum_value_sum(vector<int>& nums, int k, vector<vector<int>>& edges) {  
    int n = nums.size();  
    long long ans = 0;  
    int xor_d = 1e9;  
    int nxor_d = 1e9;  
    int cnt = 0;  
    for (int i = 0 ; i < n ; i++) {  
        int xor_val = nums[i]^k;  
        if (xor_val > nums[i]) {  
            cnt++;  
            ans += xor_val;  
            xor_d = min(xor_d , xor_val - nums[i]);  
        } else {  
            ans += nums[i];  
            nxor_d = min(nxor_d , nums[i]-xor_val);  
        }  
    }  
    if (cnt%2 == 0)  
        return ans;  
    return ans - min(xor_d, nxor_d);  
}
```

ПРИЛОЖЕНИЕ 2

Ссылка на задачу на платформе leetcode.

<https://leetcode.com/problems/find-the-maximum-sum-of-node-values>

ПРИЛОЖЕНИЕ 3

Скриншот решенной задачи на платформе.

The screenshot displays a coding platform interface. On the left, a table lists submissions with columns for Status, Language, Runtime, and Memory. The right pane shows a C++ code editor with a solution for a problem involving a graph and a dynamic programming approach.

	Status	Language	Runtime	Memory
5	Accepted 18 hours ago	C++	0 ms	126.7 MB
4	Compile Error 18 hours ago	C++	N/A	N/A
3	Accepted 18 hours ago	C++	0 ms	126.6 MB
2	Wrong Answer 19 hours ago	C++	N/A	N/A
1	Runtime Error 20 hours ago	C++	N/A	N/A

```
1 class Solution {
2 public:
3     long long maximumValueSum(vector<int>& nums, int k, vector<vector<int>>& edges) {
4         int n = nums.size();
5         long long ans = 0;
6         int xor_d = 1e9;
7         int nxor_d = 1e9;
8         int cnt = 0;
9         for (int i = 0 ; i < n ; i++) {
10             int xor_val = nums[i]*k;
11             if (xor_val > nums[i]) {
12                 cnt++;
13                 ans += xor_val;
14                 xor_d = min(xor_d , xor_val - nums[i]);
15             } else {
16                 ans += nums[i];
17                 nxor_d = min(nxor_d , nums[i]-xor_val);
18             }
19         }
20         if (cnt%2 == 0)
21             return ans;
22         return ans - min(xor_d, nxor_d);
23     }
24 };
25
```