Задача

Имеется натуральное число *n*. За один ход с ним можно произвести следующие действия:

- Вычесть единицу
- Разделить на два
- Разделить на три

При этом стоимость каждой операции – текущее значение *n*. Стоимость преобразования - суммарная стоимость всех операций в преобразовании. Вам необходимо с помощью последовательностей указанных операций преобразовать число *n* в единицу таким образом, чтобы стоимость преобразования была наименьшей. Делить можно только нацело.

Формат ввода

В первой строке строке задано 2<=n<=10^7.

Формат вывода

Выведите на первой строке искомую наименьшую стоимость. Во второй строке должна содержаться последовательность операций. Если было произведено деление на *2* или на *3*, выведите */2* (или */3*). Если же было вычитание, выведите *-1*. Все операции выводите разделяя пробелом.

Метод решения

1. **Определение состояния**:
   - Создаем массив dp, где dp[i] будет хранить минимальную стоимость преобразования числа i в 1.
   - Создаем массив op, чтобы отслеживать операции, которые привели к этому состоянию.
2. **Инициализация**:
   - Устанавливаем dp[1] = 0, так как преобразование 1 в 1 ничего не стоит.
   - Для всех других i устанавливаем dp[i] в бесконечность (или очень большое число).
3. **Рекурсия и запоминание**:
   - Для каждого числа i от 2 до n выполняем следующие действия:
     - Вычисляем стоимость операции вычитания: dp[i-1] + i и обновляем, если эта стоимость меньше текущей.
     - Если i четное, вычисляем стоимость деления на 2: dp[i/2] + i и обновляем.
     - Если i делится на 3, вычисляем стоимость деления на 3: dp[i/3] + i и обновляем.

4. **Восстановление операций**:

   - Начинаем с `n` и идем назад по массиву op, чтобы восстановить последовательность операций.

Асимптотика

По времени - O(n)

По памяти - O(2n) -> O(n)

Исходный код

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
using namespace std;
int MAX = 1e8;



int main(){
    int n;
    cin >> n;
    vector<long long> dp(n+1,MAX);
    vector<int> pos(n+1,0);
    dp[n] = n;
    for (int i = n; i>0; i--)
    {
        if (dp[i-1]>dp[i]+i-1)
        {
            dp[i-1] = dp[i]+i-1;
            pos[i-1] = i;
        }if (i%3==0 and dp[i/3]>dp[i]+i/3)
        {
```

```
            dp[i/3] = dp[i]+i/3;
            pos[i/3] = i;
        }
        if(i%2==0 and dp[i/2]>dp[i]+i/2){
            dp[i/2] = dp[i]+i/2;
            pos[i/2] = i;
        }

    }
    cout << dp[1]-1 << endl;
    string res;
    int i = 1;
    while(i!=n){
        int new_i = pos[i];
        //cout << new_i << " ";
        if (new_i == i+1){
            res += "1-";
        }else if(new_i == i*2){
            res += "2/";
        }else if(new_i == i*3){
            res += "3/";
        }
        res += " ";
        i = new_i;
    }
    res = res.substr(0,res.size()-1);
    reverse(res.begin(), res.end());
    cout << res << endl;

    return 0;
```

}

Тесты

1 - 82

2 - 1

3 - 3

4 - 10

5 - 100

6 - 1000000

Вывод

1:

202

-1 /3 /3 /3 /3

2:

0

3:

3

/3

4:

21

/2 -1 /2 -1

5:

213

/2 /2 -1 /3 /2 /2 -1

6:

2008788

/2 /2 /2 /2 /2 /2 -1 /3 /3 /2 /2 /2 -1 /3 /3 /3 /2 /2 -1