

Сортировки

Сначала разберём ошибки в дз

Задача А. Простые числа

```
#include <iostream>
using namespace std;

int main()
{
    double h;
    bool a = true;
    cin >> h;
    int n = h;
    if (h == 1 || h <= 0 || h - n > 0) a = false;
    for (int i = 2; i*2 <= n; ++i) {
        if (n % i == 0) a = false;
    }
    if (a) cout << "Yes";
    else cout << "No";
}
```

Задача А. Простые числа

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int h,n=0;cin >> h;
    for(int i=2;i <= sqrt(h);i++){
        if(h%i==0){
            n++;
        }
    }
    if(h==1){
        cout << "No";
    }
    else if(n==0){
        cout << "Yes";
    }
    else{
        cout << "No";
    }
}
```

Задача В.

```
#include <iostream>
using namespace std;
int main() {
    int n, i = 2; cin >> n;
    while (n > 1) {
        int k = 0;
        while (n % i == 0) {
            k++;
            n = n / i;
        }
        if (k > 0) {
            cout << i;
            if (k > 1) {
                cout << "A" << k;
            }
            if (n > 1){
                cout << "*";
            }
        }
        if (i == 2) {
            i++;
        }
        else {
            i += 2;
        }
    }
    cout << endl;
    cin.get(); cin.get();
    return 0;
}
```

```
3. using namespace std;
4.
5. int main() {
6.     int n, i = 2;
7.     cin >> n;
8.     while (n > 1) {
9.         int k = 0;
10.
11.         while (n % i == 0) {
12.             k++;
13.             n = n / i;
14.         }
15.
16.         if (k > 0) {
17.             cout << i;
18.             if (k > 1) {
19.                 cout << "A" << k;
20.             }
21.             if (n > 1){
22.                 cout << "*";
23.             }
24.         }
25.
26.         if (i == 2) {
27.             i++;
28.         }
29.         else {
30.             i += 2;
31.         }
32.     }
33.     cout << endl;
34.     cin.get(); cin.get();
35.     return 0;
36. }
```

Задача В.

```
3. using namespace std;
4.
5. int main() {
6.     int n, i = 2;
7.     cin >> n;
8.     while (n > 1) {
9.         int k = 0;
10.
11.         while (n % i == 0) {
12.             k++;
13.             n = n / i;
14.         }
15.
16.         if (k > 0) {
17.             cout << i;
18.             if (k > 1) {
19.                 cout << "^" << k;
20.             }
21.             if (n > 1){
22.                 cout << "*";
23.             }
24.         }
25.
26.         if (i == 2) {
27.             i++;
28.         }
29.         else {
30.             i += 2;
31.         }
32.     }
33.     cout << endl;
34.     cin.get(); cin.get();
35.     return 0;
36. }
```

Задача В.

```
void dec(int n) {  
    int i = 2;  
    int deg = 0;  
    bool first = 1;  
    int res = n;  
  
    if (is_prime(res)) {  
        cout << res;  
    }  
    else {  
        while (n != 1) {  
            if (is_prime(i) && n % i == 0) {  
                while (n % i == 0) {  
                    n /= i;  
                    deg++;  
                }  
                if (deg == 1) {  
                    cout << i;  
                }  
                else if (deg > 1) {  
                    cout << i << "^" << deg;  
                }  
                if (n != 1) {  
                    cout << "*";  
                }  
            }  
            deg = 0;  
            if (i == 2) {  
                i++;  
            }  
            else {  
                i += 2;  
            }  
        }  
    }  
}
```

Задача В.

```
#include <iostream>
#include <vector>
#include <math.h>
using namespace std;

int main()
{
    int s, p = 0;
    vector<int> a(0);
    vector<int> b(0);
    cin >> s;
    int n = s;
    if (s == 2 || s == 3) cout << s;
    for (int i = 2; i <= n/2; ++i) {
        p = 0;
        while (s % i == 0) {
            s /= i;
            ++p;
        }
        if (p > 0) {
            b.push_back(p);
            a.push_back(i);
        }
    }
    for (int i = 0; i < a.size(); ++i) {
        if (i > 0) cout << '*';
        if (b[i] > 1) cout << a[i] << '^' << b[i];
        else cout << a[i];
    }
    return 0;
}
```


Ну всё-всё, теперь сортировки

На самом деле, некоторые из вас, может и не задумывались об этом, но писали свою первую сортировку

На самом деле, некоторые из вас, может и не задумывались об этом, но писали свою первую сортировку

Е. Кадиллак

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Однажды музыкальный исполнитель Моргенштерн записывал клип "Кадиллак". Этот клип очень понравился Пете. Он настолько ему понравился, что слушал его в колонках во дворе. Но все бы ничего, если бы Петя не заметил одну особенность клипа: откуда у машин такие номера?. При чем машины были из разных регионов России. У Петя появился вопрос: а в каком регионе России больше всего любят Моргенштерна? Для этого он решил узнать регион, из которого приехало больше всего машин. Может это не будет соответствовать действительности, но для Пети это правда важно!

Входные данные

На первой строке ввода находится два числа n и k ($1 \leq n, k \leq 10^6$) - кол-во машин в клипе и кол-во регионов в России. На второй строке ввода находится n чисел - регионы каждой из машин. Если reg_i регион i -й машины, то $1 \leq reg_i \leq k$.

Выходные данные

Выведите ответ на вопрос Пети: из какого региона машин больше всего? Вы должны вывести одно число. Если ответов несколько - вывести наименьший

Примеры

входные данные	Скопировать
10 5 1 4 2 3 5 1 1 5 1 3	
выходные данные	Скопировать
1	

входные данные	Скопировать
10 7 3 6 5 4 3 2 4 3 1 3	
выходные данные	Скопировать
3	

Примечание

Такая задача (только несколько в иной формулировке) может встретиться вам на ЕГЭ по информатике

Помните такое было в дз2?

Сортировка подсчётом

Сортировка подсчётом

- У нас есть n элементов, которые не больше, чем k

Сортировка подсчётом

- У нас есть n элементов, которые не больше, чем k
- Тогда просто заведём вектор размером $k + 1$ (назовём его x)

Сортировка подсчётом

- У нас есть n элементов, которые не больше, чем k
- Тогда просто заведём вектор размером $k + 1$ (назовём его x)
- Будем идти по массиву, допустим нам встретилось число q , тогда в $x[q]$ будем добавлять единичку

Сортировка подсчётом

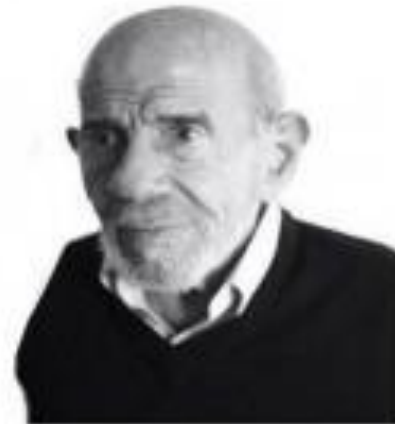
- У нас есть n элементов, которые не больше, чем k
- Тогда просто заведём вектор размером $k + 1$ (назовём его x)
- Будем идти по массиву, допустим нам встретилось число q , тогда в $x[q]$ будем добавлять единичку
- Потом мы можем составить исходный массив...

Сортировка подсчётом

- У нас есть n элементов, которые не больше, чем k
- Тогда просто заведём вектор размером $k + 1$ (назовём его x)
- Будем идти по массиву, допустим нам встретилось число q , тогда в $x[q]$ будем добавлять единичку
- Потом мы можем составить исходный массив...

Как?

— Жак Фреско —



Сортировка подсчётом

Сортировка подсчётом

- Самый простой (но не совсем правильный с точки зрения идеи)
пример – выборы в президенты

Сортировка подсчётом

- Самый простой (но не совсем правильный с точки зрения идеи) пример – выборы в президенты
- Пример бытовой – определить, у кого из учеников больше всего оценок 5

Сортировка подсчётом

- Самый простой (но не совсем правильный с точки зрения идеи) пример – выборы в президенты
- Пример бытовой – определить, у кого из учеников больше всего оценок 5
- Ещё ловите – менеджеру нужно определить, какой суп оставить в меню (а определяет он это не по выручке, а по кол-ву заказов супов)

Давайте при вас решу задачку Кадиллак

А ещё покажу, как получить
отсортированный вектор

Какая асимптотика?

Какая асимптотика?

- $O(|q| + n)$, где $|q|$ - размер диапазона получаемых чисел, а n – количество чисел

Сортировка пузырьком

Сортировка пузырьком

Сортировка пузырьком

- Какова идея? Просто менять местами два элемента, если левый элемент больше правого (и так много раз)

Сортировка пузырьком

- Какова идея? Просто менять местами два элемента, если левый элемент больше правого (и так много раз)
- Нам придётся n раз сделать вот такую штуку:

Сортировка пузырьком

- Какова идея? Просто менять местами два элемента, если левый элемент больше правого (и так много раз)
- Нам придётся n раз сделать вот такую штуку:

Пройтись по массиву, и если можно поменять элементы местами - менять

Сортировка пузырьком

- Какова идея? Просто менять местами два элемента, если левый элемент больше правого (и так много раз)
- Нам придётся $n - 1$ раз сделать вот такую штуку:

Пройтись по массиву, и если можно поменять элементы местами - менять

```
for (int q = 1; i < q.size(); ++q) {  
    for (int i = 1; i < x.size(); ++i) {  
        if (x[i - 1] > x[i]) {  
            swap(x[i - 1], x[i]);  
        }  
    }  
}
```

Возникают вопросы

Возникают вопросы

- Какова асимптотика сортировки

Возникают вопросы

- Какова асимптотика сортировки
- Почему это работает?

Асимптотика сортировки пузырьком

Асимптотика сортировки пузырьком

- Мы $n - 1$ раз проходимся по всему массиву

Асимптотика сортировки пузырьком

- Мы $n - 1$ раз проходимся по всему массиву
- Один проход работает за $O(n)$

Асимптотика сортировки пузырьком

- Мы $n - 1$ раз проходимся по всему массиву
- Один проход работает за $O(n)$
- Тогда итоговая асимптотика = $O(n^2)$

Для следующего алгоритма нам надо
научиться кое-какой штуке

Для следующего алгоритма нам надо
научиться кое-какой штуке

- Допустим уже есть какой-то заполненный вектор x

Для следующего алгоритма нам надо научиться кое-какой штуке

- Допустим уже есть какой-то заполненный вектор x
- К нам поступают запросы вида «вставить на k -ю позицию элемент b , а элементы справа – просто сдвинуть вправо» (размер вектора, таким образом, увеличивается на 1)

Для следующего алгоритма нам надо научиться кое-какой штуке

- Допустим уже есть какой-то заполненный вектор x
- К нам поступают запросы вида «вставить на k -ю позицию элемент b , а элементы справа – просто сдвинуть вправо» (размер вектора, таким образом, увеличивается на 1)
- Давайте посмотрим визуально, что хотят

Было

6	4	16	32	2	432	1234
0	1	2	3	4	5	6

Было

6	4	16	32	2	432	1234
0	1	2	3	4	5	6

Было

6	4	16	32	2	432	1234
0	1	2	3	4	5	6

6	4	16	32	2	432	1234
0	1	2	3	4	5	6



На позицию 3
вставить элемент 41

Было

6	4	16	32	2	432	1234
0	1	2	3	4	5	6

6	4	16	32	2	432	1234
0	1	2	3	4	5	6



На позицию 3
вставить элемент 41

Вот так это будет
выглядеть

6	4	16	41	32	2	432	1234
0	1	2	3	4	5	6	7

Асимптотика?

- Асимптотика одной вставки будет $O(n)$
- Почему?
- Нам нужно будет сдвинуть все элементы справа. В худшем случае, мы будем вставлять элемент нулевым и нужно будет двигать все элементы

Предположим ситуацию

Предположим ситуацию

- У вас уже есть отсортированный вектор x

Предположим ситуацию

- У вас уже есть отсортированный вектор x
- Вас просят добавить в вектор число k так, чтобы вектор остался отсортированным

Предположим ситуацию

- У вас уже есть отсортированный вектор x
- Вас просят добавить в вектор число k так, чтобы вектор остался отсортированным
- Как вы это будете делать?

Предположим ситуацию

Сортировка вставками

- В самом начале у нас есть пустой вектор

Сортировка вставками

- В самом начале у нас есть пустой вектор
- Мы будем по одному вставлять (по-умному) в него числа

Сортировка вставками

- В самом начале у нас есть пустой вектор
- Мы будем по одному вставлять (по-умному) в него числа
- В итоге, после вставки последнего числа, все элементы в векторе отсортированы

Сортировка вставками

- В самом начале у нас есть пустой вектор
- Мы будем по одному вставлять (по-умному) в него числа
- В итоге, после вставки последнего числа, все элементы в векторе отсортированы
- Идеи по реализации: было бы неплохо написать функцию вставки, в неё по ссылке передавать вектор и элемент, который нужно вставить

И снова, асимптотика?

И снова, асимптотика?

- Нам нужно будет произвести n вставок

И снова, асимптотика?

- Нам нужно будет произвести n вставок
- Каждая вставка работает за $O(n)$

И снова, асимптотика?

- Нам нужно будет произвести n вставок
- Каждая вставка работает за $O(n)$
- Тогда n вставок отработают за $O(n^2)$

Если успеем, разберём ещё одну сортировку (и про неё я напишу в презентации)