

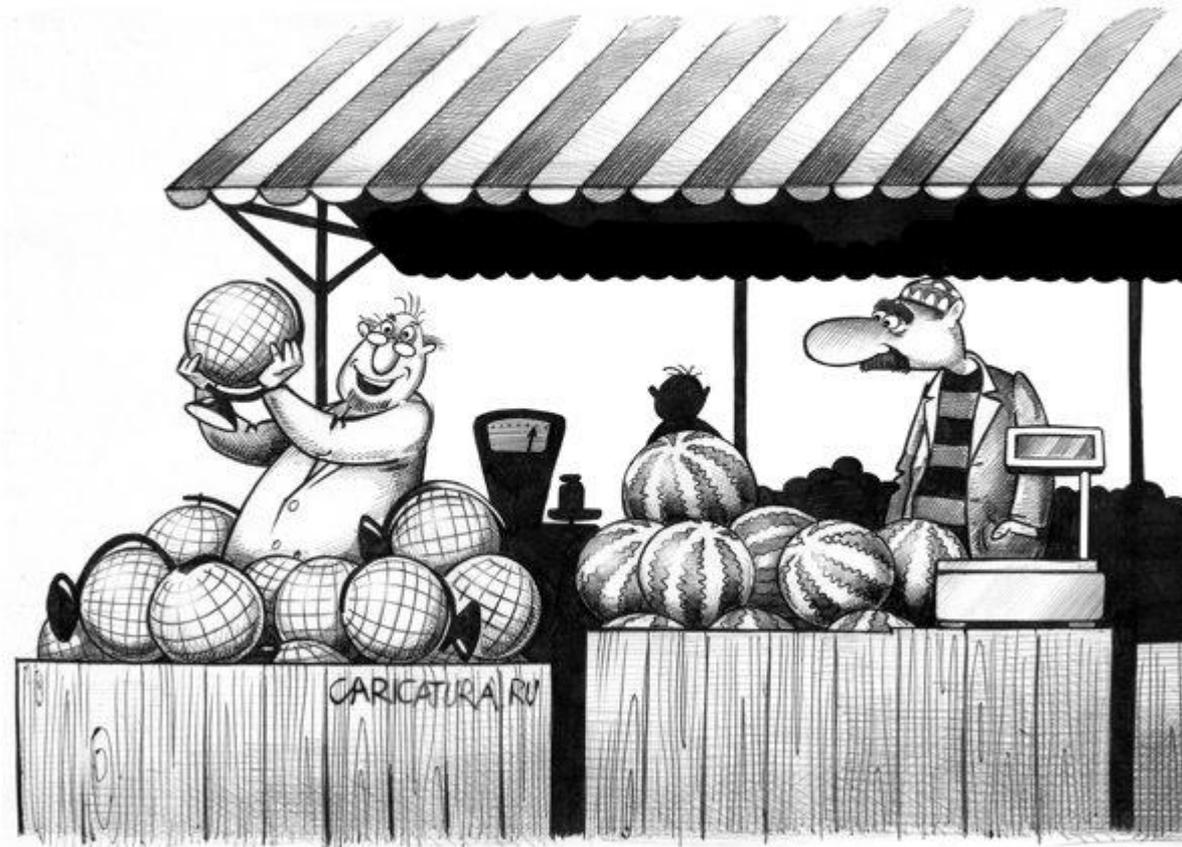
# Школьный этап Всероссийской олимпиады школьников по информатике 10-11 класс

Разбор задач

Липецк, 2016

# Задача А

# РЫНОК



# Условие задачи

Сегодня Максим решил сходить на рынок, чтобы купить помидоры. На рынке в ряд стоят  $N$  продавцов, каждый из которых продаёт помидоры. Также Максиму известны  $N-1$  различных чисел  $a_i$  – суммарное количество помидоров, продающихся у всех продавцов, стоящих правее некоторого продавца.

Помогите Максиму определить, сколько помидоров продаёт самый правый продавец.

# Как решать?

- Достаточно понять, что чем меньше  $a_i$ , тем правее стоит продавец.
- Таким образом, справа от предпоследнего продавца будет продаваться  $\min_{i=1\dots N} a_i$  помидоров.
- А так как справа от предпоследнего продавца стоит только один – последний продавец, то у него будет именно продаваться  $\min_{i=1\dots N} a_i$  помидоров.
- То есть достаточно найти минимальное число в массиве – это можно сделать за один проход по нему.

# Фрагмент программы

```
int min = a[0];  
for (int i = 0; i < n; ++i) {  
    if (a[i] < min) {  
        min = a[i];  
    }  
}
```

# Задача В

## Часы



# Условие задачи

Однажды Максим проснулся и взглянул на часы. Часы могут показывать время в 12- или 24-часовом формате НН:ММ. В 12-часовом формате часы изменяются в пределах от 1 до 12, а в 24-часовом формате – от 0 до 23. Минуты в обоих форматах изменяются от 0 до 59. Максиму известно, в каком формате сейчас показывают часы (12- или 24-часовой), а также он видит время в формате НН:ММ, которое показывают часы. Иногда часы Максима ломаются и показывают время неправильно (например, 99:99). Максиму известно, в каком формате (12- или 24-часовом) сейчас показывают время часы.

Помогите Максиму понять, правильно ли часы показывают время.

# Как решать?

- Если часы показывают время в 12-часовом формате, достаточно проверить, что количество часов лежит в промежутке от 1 до 12 включительно, а количество минут – в промежутке от 0 до 59 включительно.
- Если же часы показывают время в 24-часовом формате, то нужно проверить, что количество часов лежит в промежутке от 0 до 23 включительно, а количество минут – в промежутке от 0 до 59 включительно.

# Фрагмент программы

```
if (type == 12) {  
    if (h >= 1 && h <= 12 && m >= 0 && m <= 59) {  
        cout << "Right" << endl;  
    } else {  
        cout << "Wrong" << endl;  
    }  
} else {  
    if (h >= 0 && h <= 23 && m >= 0 && m <= 59) {  
        cout << "Right" << endl;  
    } else {  
        cout << "Wrong" << endl;  
    }  
}
```

# Задача С

## Номера чеков



# Условие задачи

Однажды Максиму надоело чинить свои часы и он пошёл в банк, чтобы взять денег, а потом купить себе новые часы. В банке каждый клиент получает чек с номером окна, в котором его должны обслужить. Всего в банке  $N$  окон с номерами от 1 до  $N$ . Система иногда работает со сбоями, и не полностью пропечатывает номера чеков. Максиму выдали чек, на котором было напечатано число  $K$ . Цифры на чеке выглядят как на картинке. Каждая цифра состоит из 7 частей, каждая из которых может либо пропечататься, либо нет. Если номер окна меньше 10, то печатается ведущий ноль. Ему стало интересно, номера скольких окон могли так пропечататься.

# Как решать?

- Достаточно для каждой из цифр от 0 до 9 хранить список цифр, которые могут получиться из текущей цифры путём добавления какой-либо из 7 частей индикатора.
- Например, из 0 могут получиться цифры: 0, 8. А из 1 – цифры: 0, 1, 3, 4, 7, 8, 9.
- Далее перебираем всевозможные варианты чисел, которые могут получиться из исходного числа, проверяем, что они больше 1 и меньше  $N$ , и считаем количество чисел, которые подошли под это условие.

# Фрагмент программы

```
int answer = 0;
for (int i = 1; i <= n; ++i) {
    if (good(k / 10, i / 10) && good(k % 10, i % 10))
    {
        ++answer;
    }
}
```

# Задача D

## Путешествие



© www.ClipProject.info

# Условие задачи

Максим очень любит путешествовать, поэтому сегодня он решил отправиться в горы. Горы представляют собой  $N$  холмов, каждый из которых имеет высоту  $a_i$ . Холмы нумеруются слева направо числами от 1 до  $N$ . Максим передвигается по холмам слева направо. Помогите для каждого холма с номером  $i$  определить минимальный номер холма  $j$ , чтобы выполнялись следующие условия:  $i < j, a_i > a_j$ . Если такого  $j$  не существует, выведите  $-1$ .

# Как решать?

## Наивное решение

- Достаточно понять, что чем меньше  $a_i$ , тем правее стоит продавец.
- Таким образом, справа от предпоследнего продавца будет продаваться  $\min_{i=1\dots N} a_i$  помидоров.
- А так как справа от предпоследнего продавца стоит только один – последний продавец, то у него будет именно продаваться  $\min_{i=1\dots N} a_i$  помидоров.
- То есть достаточно найти минимальное число в массиве – это можно сделать за один проход по нему.

# Фрагмент программы

```
vector<int> ans(n);
for (int i = 0; i < n; ++i) {
    int index = -1;
    for (int j = i + 1; j < n; ++j) {
        if (a[j] < a[i] && index == -1) {
            index = j;
            break;
        }
    }
    ans[i] = index;
}
```

# Как решать? Оптимальное решение

- В данной задаче необходимо было для каждого числа найти ближайшее справа, меньшее число. Это делается при помощи стека.
- Идем по исходному массиву с конца, и будем формировать стек.
- Для каждого элемента массива делаем следующее: до тех пор, пока стек не пуст, и в вершине стека лежит число, большее текущего, то удаляем число из стека.
- Если после проделанных операций стек окажется пустым, то ответ для текущего числа равен  $-1$ , в противном случае он равен числу, лежащему на вершине стека.
- После сохранения ответа, добавляем в стек текущее число, и переходим к следующему элементу.
- Так как в ответе нужно вывести не сами ближайшее справа числа, меньше данного, а их индексы, то в стеке можно хранить пары чисел вида (число; индекс).
- Данное решение работает за линейное время  $O(N)$ .

# Фрагмент программы

```
vector<int> ans(n, -1);
stack<int> st;
for (int i = n - 1; i >= 0; --i) {
    while (!st.empty() && a[st.top()] >= a[i]) {
        st.pop();
    }
    if (!st.empty()) {
        ans[i] = st.top();
    }
    st.push(i);
}
```



Спасибо за внимание!  
Вопросы?