

*Кучеренко Демид Сергеевич,
Ульянцев Владимир Игоревич*

ЗАДАЧА «ГОНКИ НА ПОДАХ»

Этой статьей мы продолжаем цикл публикаций олимпиадных задач для школьников по информатике. Решение таких задач и изучение разборов поможет Вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике.

В этой статье рассматривается задача «Гонки на подах», которая предлагалась на четвертой интернет-олимпиаде базового уровня сезона 2012–2013. Интернет-олимпиады по информатике базового уровня проводятся Санкт-Петербургским национальным исследовательским университетом информационных технологий, механики и оптики. Сайт этих олимпиад находится по адресу <http://neerc.ifmo.ru/school/io>.

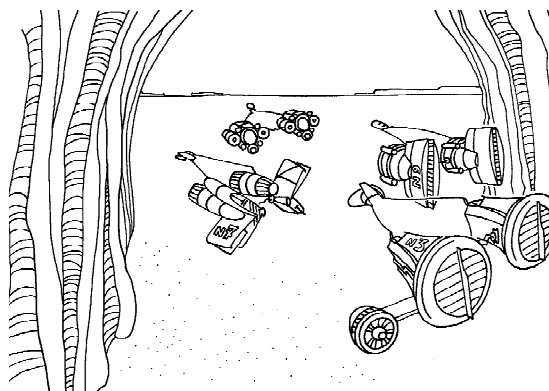
УСЛОВИЕ ЗАДАЧИ

Гонки на подах – популярный вид развлечений на планете Татуин. Удовольствие от них способны получить и участники, и зрители, и букмекерские конторы, принимающие ставки на результат очередной гонки. При этом владельцы букмекерских контор периодически совершают некоторые незаконные действия, чтобы повлиять на итоговые позиции гонщиков. Например, можно испортить тормозную систему в некоторых подах, после чего гонщику придется ехать существенно медленнее обычной скорости

ради сохранения контроля гонщика над машиной.

Выяснилось, что в последней гонке участвовали n гонщиков. У каждого из гонщиков на машине был написан номер – число от 1 до n , номера всех гонщиков различались. Также известно, что владельцы одной из букмекерских контор испортили тормозную систему во всех машинах, номера которых превосходили некоторое число k . Любая машина с испорченной тормозной системой будет ехать медленнее, чем любая машина с исправными тормозами. Соответственно, в протоколе с результатами гонки у любой машины с номером большим, чем k , место будет также больше, чем k .

Вы проводите расследование этого неприятного инцидента. В качестве первого шага расследования вы решили найти все возможные значения числа k , изучая только результаты гонки.



Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \leq n \leq 10^5$) – количество гонщиков, участвовавших в соревновании. Вторая строка содержит n попарно различных чисел a_i ($1 \leq a_i \leq n$) – протокол с результатами гонки, где a_i – номер машины, которая заняла i -ое место.

Формат выходного файла

В первой строке выходного файла выведите одно целое число c – количество возможных значений числа k . В следующей строке выведите c натуральных чисел, разделенных пробелами – возможные значения числа k . Все числа во второй строке должны быть различны и не должны превосходить n .

Числа во второй строке должны быть упорядочены по возрастанию.

Примеры входных и выходных данных

race.in	race.out
6 2 1 4 3 6 5	3 2 4 6

РАЗБОР ЗАДАЧИ

Формализуем поставленную задачу. Во входном файле задана перестановка из n чисел. Необходимо найти все элементы перестановки, для которых выполняется следующее правило: каждый элемент, который находится правее данного, больше данного элемента и всех, находящихся левее данного. Действительно, если можно было сломать только те машины, номер которых больше k , то эти машины и займут места с $(k + 1)$ по n , а их номера будут больше всех остальных.

Пусть наш элемент a_i на позиции i соответствует этому правилу. Так как все элементы $a_{i+1} \dots a_n$ больше всех элементов $a_1 \dots a_i$, то максимум первых i элементов перестановки равен i (количество элементов перестановки, больших чем i , равно $n - i$, а количество мест, куда нужно разместить эти элементы, тоже $n - i$).

Для определения позиций, удовлетворяющих нашему правилу, будем просто поддерживать максимум на префиксе массива.

Листинг 1. Реализация описанного решения

```
uses
  Math;
var
  i, n, maximum, size : longint;
  a, ans : array [1..100000] of longint;
begin
  reset(input, "race.in");
  rewrite(output, "race.out");
  read(n);
  for i := 1 to n do
    read(a[i]);
  maximum := 0;
  for i := 1 to n do begin
    maximum := max(maximum, a[i]);
    if (maximum = i) then begin
      inc(size);
      ans[size] := i;
    end;
  end;
  writeln(size);
  for i := 1 to size do
    write(ans[i], " ");
end.
```

Если в какой-то момент текущий максимум равен i , добавим данное i в ответ.

Поскольку мы проходим массив в порядке увеличения индекса, то добавлять эти индексы в массив мы будем тоже в порядке увеличения, как и требуется в условии задачи. В листинге 1 приведена реализация описанного решения на языке *Pascal*.

Время работы приведенного решения составляет $O(n)$. Заметим, что все решения, время работы которых составляет $O(n^2)$, не удовлетворяют ограничению по времени, которое на олимпиадах обычно составляет две секунды работы персонального компьютера.



Наши авторы, 2013.
Our authors, 2013.

*Кучеренко Демид Сергеевич,
студент третьего курса кафедры
«Компьютерные технологии»
НИУ ИТМО, член жюри Интернет-
олимпиад по информатике,*

*Ульянцев Владимир Игоревич,
аспирант кафедры «Компьютерные
технологии» НИУ ИТМО,
член жюри Интернет-олимпиад
по информатике.*