Школа современного программирования

ГОТОВИМСЯ К ОЛИМПИАДАМ ПО ИНФОРМАТИКЕ

Шовкопляс Григорий Филиппович,
Ульянцев Владимир Игоревич

ЗАДАЧА «ПОЧИНКА ЗАБОРА»

Этой статьей мы начинаем цикл публикаций олимпиадных задач по информатике для школьников 2014 года. Решение таких задач и изучение их разборов поможет повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике. В этой статье рассматривается задача «Починка забора», которая предлагалась на Пятнадцатой Всероссийской командной олимпиаде школьников по программированию. Материалы этой олимпиады можно найти на сайте http://neerc.ifmo.ru/school/russia-team/.

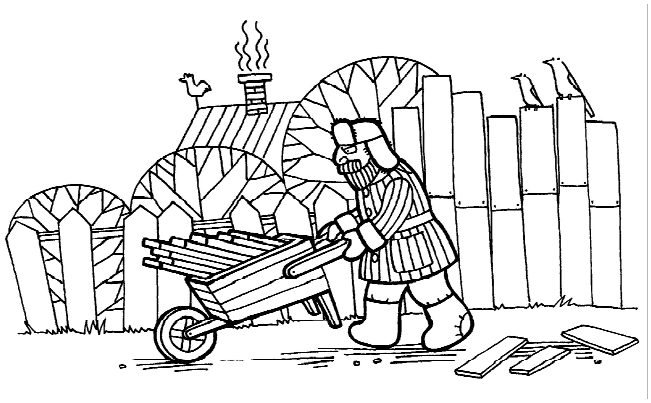
УСЛОВИЕ ЗАДАЧИ

Дед Афанасий решил подлатать забор вокруг своего деревенского участка. В прошлом году он как раз строил сарай, так что доски для ремонта у него остались. Забор состоит из n сегментов, каждый из которых представляет собой доску высотой a_i. У деда есть тележка, на которой лежит стопка из m досок, про каждую из которых известна ее длина b_i.

Дед Афанасий идет вдоль забора и катит перед собой тележку с досками. Если он хочет увеличить высоту текущего сегмента, он может взять доску с тележки и прибить ее сверху. Тогда новая высота сегмента будет равна сумме изначальной высоты сегмента и длины прибитой доски. При этом дед не хочет прибивать больше одной доски к каждому сегменту забора, чтобы сохранить его прочность.

Собираясь увеличить высоту сегмента забора, Афанасий поступает следующим образом. Он либо использует для увеличения сегмента верхнюю доску с тележки, либо выкидывает одну или несколько верхних досок с тележки и использует следующую за ними доску. Силы у Афанасия уже не те, поэтому он никогда не возвращается назад вдоль забора и никогда не подбирает выкинутые ранее доски.

Перед началом работы дед задумался, какую максимальную высоту может иметь забор после починки. Высотой забора Афанасий считает высоту самого низкого сегмента забора. Помогите деду Афанасию узнать, какую максимальную высоту забора он сможет получить.



Формат входного файла

В первой строке входного файла задано число n ($1 \le n \le 10^5$) – количество сегментов в заборе. Во второй строке содержится n целых чисел a_i ($1 \le a_i \le 10^8$) – высота сегмента забора с номером i.

В третьей строке – число m ($1 \le m \le 10^5$) – количество досок на тележке. В четвертой строке через пробел содержится m целых чисел b_i ($1 \le b_i \le 10^8$) – длины досок на тележке, начиная с верхних.

Формат выходного файла

В первую строку выходного файла выведите два целых числа h и k – максимальную возможную высоту забора и количество досок, которые деду следует использовать при починке. В следующих k строках выведите по два целых числа x_i и y_i, которые означают, что к x_i-му сегменту забора деду следует прибить доску с номером y_i.

Если существует несколько способов починить забор требуемым образом, выведите любой из них.

Примеры входных и выходных данных

wall.in	wall.out
3 10 5 10 1 5	10 1 2 1
6 2 5 4 1 7 5 7 2 3 1 3 2 4 6	5 3 1 2 3 4 4 7

РАЗБОР ЗАДАЧИ

Чтобы решить данную задачу, нужно найти две последовательности индексов x_i и y_i длины k, такие что: $1 \le x_1 < x_2 < ... < x_k \le n$ и $1 \le y_1 < y_2 < ... < y_k \le m$, чтобы после того, как дед прибьет к каждому x_i-му сегменту y_i-ю доску, высота нового забора была максимальной. Будем решать эту задачу при помощи «бинарного поиска по ответу»: перебирая высоту h бинарным поиском, будем проверять, можем ли мы ее достичь или нет. Итоговый ответ получится по окончанию работы бинарного поиска.

В листинге 1 приведена реализация бинарного поиска по ответу на языке Pascal. В реализации будем придерживаться следующего инварианта: максимальная высота h, которую Афанасий может получить, лежит в промежутке от l включая, до r не включая r.

Функция `check` выполняет проверку, можем ли мы починить забор таким образом, чтобы его высота была хотя бы h (листинг 2). В ней мы моделируем процесс починки забора, проходя мимо каждого сегмента и, если его высота меньше h, ищем в стопке доску, которую нужно прибить, чтобы новая высота была хотя бы h. Если в какой-то момент такую доску мы не находим, то, очевидно, ответ на проверку негативный.

Осталось только восстановить ответ, что можно сделать, промоделировав починку еще раз аналогично функции `check` с некоторыми отличиями: во-первых мы точно знаем, что такую высоту можно получить, а во-вторых нужно вывести способ починки забора.

В листинге 3 приведена реализация функции `write_ans`, которая получает в качестве аргумента максимально возможную

Листинг 1. Бинарный поиск по высоте забора

```
l := 0;
r := 1000000000;
while l < r - 1 do begin
  cur := (l + r) div 2;
  if check(cur) then
    l := cur
  else
    r := cur;
end;
```

Листинг 2. Проверка возможности достижения высоты *h*

```
function check(h: longint): boolean;
var
  i, j: longint;
  is_good: boolean;
begin
  is_good := true;
  i := 0;
  j := 1;
  while is_good do begin
    inc(i);
    if i > n then
      break;
    if a[i] >= h then
      continue;
    while j <= m do begin
      if a[i] + b[j] >= h then
        break;
      inc(j);
    end;
    if j > m then
      is_good := false;
    inc(j);
  end;
  check := is_good;
end;
```

Листинг 3. Восстановление и вывод ответа

```
procedure write_ans(h: longint);
var
  i, j, k: longint;
begin
  i := 0;
  j := 1;
  k := 1;
  while i < n do begin
    inc(i);
    if a[i] >= h then
      continue;
    while j <= m do begin
      if a[i] + b[j] >= h then
        break;
      inc(j);
    end;
    x[k] := i;
    y[k] := j;
    inc(k);
    inc(j);
  end;
  writeln(h, " ", k - 1);
  for i := 1 to k - 1 do
    writeln(x[i], " ", y[i]);
end;
```

высоту, восстанавливает ответ и выводит его в требуемом формате.

Заметим, что по приведенному коду несложно оценить время его работы: суммарно программа выполнит $O(N \times \log 10^9)$ операций, и ее время выполнения при данных в условии ограничениях уложится в ограничение на время работы программы.

Шовкопляс Григорий Филиппович,
студент второго 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.