Разбор задачи «G. Операционные системы»

Автор задачи и разбора — К. Абакумов

В версии для лиги B количество установленных операционных систем не превосходит 1 000. Поэтому можно применить простой алгоритм: для каждой из систем будем проверять, пересекается ли она с системами, установленными позже. Если нет, тогда она «выжила» и надо прибавить к ответу 1.

В лиге A количество операционных систем не превосходит 100 000. Операционная система останется выжившей, если между ее первым и последним сектором включительно нет систем, установленных позже. Для каждой будем хранить флаг, в котором будет записано *true*, если операционная система может быть выжившей и $false$, если уже не может. Запишем все события (начальный и конечный сектор каждой из операционной систем) в массив и отсортируем их по двум параметрам: номеру сектора и, в случае равенства, по типу события — сначала идут события, отвечающие за начало системы, затем отвечающие за конец. Мы будем пробегаться подряд по отсортированным событиям жесткого диска и хранить номера всех операционных систем, уже начавшихся, но ещё не закончившихся (то есть было встречено начало системы, но ещё не было её конца). При инициализации все системы выжившие, и никакая система ещё не началась. Если текущее событие событие — начало системы, то мы добавляем ее в список, если на событие удаления — то удаляем из него.

Заметим, что при добавлении в список системы с номером P все системы из списка с номерами, меньшими P, уже не могут быть выжившими. В реализации нужно пометить как $false$ либо её (если P меньше максимального в списке), либо максимальную систему в списке (в противном случае), ведь все остальные элементы списка были помечены как $false$ еще на предыдущих шагах. При удалении системы нужно проверить её флаг: если он *true*, то она гарантированно остаётся, поэтому надо прибавить к ответу 1.

Описанное решение работает за $O(n^2)$. Чтобы добиться оценки $O(n \log n)$, вместо списка достаточно использовать кучу, которая позволяет быстро искать максимум, добавлять и удалять элементы. Её реализацию можно найти, например, на странице

`http://informatics.mccme.ru/moodle/mod/book/view.php?id=454&chapterid=325`

В корне дерева будем хранить систему с максимальным номером. При добавлении нового элемента запускается `sift_up`, в которой все системы, оказавшиеся на пути новой до корня, помечаются как $false$. Поэтому либо добавляемая система сразу попадёт в корень кучи, а флаг бывшего корня станет $false$, либо она сама будет помечена не выжившей.

Итак, все элементы в куче кроме корня всегда имеют флаг выживания, равный $false$, так как в текущий момент времени для каждой системы кроме корневой имеется как минимум ещё одна с большим номером, с которой она пересекается (корень кучи).

Вот пример реализации алгоритма:

```
const
  MaxN = 100000;

var
  heap, place : array[1 .. MaxN] of integer; //place[i] - место i-ой системы в куче
  action, sys, disk : array[1 .. 2 * MaxN] of integer;
```

```
  //action[i] - тип события, sys[i] - номер системы в i-ом событии,
  //disk[i] - номер сектора
  survived : array[1 .. MaxN] of boolean;
  //survived[i] - флаг, обозначающий может ли еще система выжить или уже нет
  n, m, i, hs, k : integer;

procedure sift_up(n : integer); //"Всплывание" n-ого элемента кучи
begin
  if n = 1 then exit;
  if heap[n] > heap[n div 2] then begin
    ... // поменять n и n div 2
    sift_up(n div 2);
  end;
  survived[heap[n]] := false
end;
procedure sift_down(s : integer); //"Утопление" s-ого элемента в куче
procedure del(n : integer); //Удаление n-ого элемента из кучи
procedure ins(x : integer); //Добавление элемента x в кучу
procedure sort(l, r: integer); //Сортировка событий
begin
  reset(input, 'g.in');
  rewrite(output, 'g.out');
  read(n, n);
  m := 0;
  //Считываем информацию об операционных системах и заполняем массив событий
  for i := 1 to n do begin
    inc(m);
    read(disk[m]);
    sys[m] := i;
    action[m] := 1;
    inc(m);
    read(disk[m]);
    sys[m] := i;
    action[m] := -1;
  end;
  if m <> 0 then sort(1, m);
  fillchar(survived, sizeof(survived), true);
  hs := 0;k := 0;
  for i := 1 to m do
    if action[i] > 0 then begin
      ins(sys[i]);
    end else begin
      if survived[sys[i]] then inc(k);
      del(place[sys[i]]);
    end;
  writeln(k);
  close(input);close(output)
end.
```