



Хана

Жиан-Жиа ижил хэмжээтэй тоосгонуудыг давхарлан хана барьж байгаа. Уг хана нь зүүнээс баруун тийш 0-ээс $n - 1$ хүртэл дугаарлагдсан n ширхэг баганануудаас тогтоно. Багананууд нь өөр өөр өндөртэй байж болно. Багананы өндөр гэдэг нь тухайн багана дахь тоосгоны тоог хэлнэ.

Жиан-Жиа ханыг дараах аргаар барина. Эхлээд бүх баганд ямар ч тоосго байхгүй. Тэгээд Жиан-Жиа тоосго нэмэх эсвэл хасах гэсэн k удаагийн үйлдэл хийнэ. k үйлдэл дуусахад хана барих процесс дуусна. Жиан-Жиад үйлдэл бүрд, өөрчлөлт хийх баганануудыг агуулсан завсар болон өндөр h өгөгдөх бөгөөд тэрээр доорх үйлдлийг хийнэ:

- нэмэх үйлдэлд, Жиан-Жиа өгөгдсөн завсарт h -ээс цөөн тоосготой баганануудад тоосго нэмэх байдлаар тэдгээр баганануудыг яг h ширхэг тоосготой болгоно. Тэрээр h -аас олон эсвэл тэнцүү тооны тоосготой баганануудад үйлдэл хийхгүй.
- хасах үйлдэлд, Жиан-Жиа өгөгдсөн завсарт h -ээс олон тоосготой баганануудаас тоосго хасах байдлаар тэдгээр багануудыг яг h ширхэг тоосготой болгоно. Тэрээр h -аас цөөн эсвэл тэнцүү тооны тоосготой багануудад үйлдэл хийхгүй.

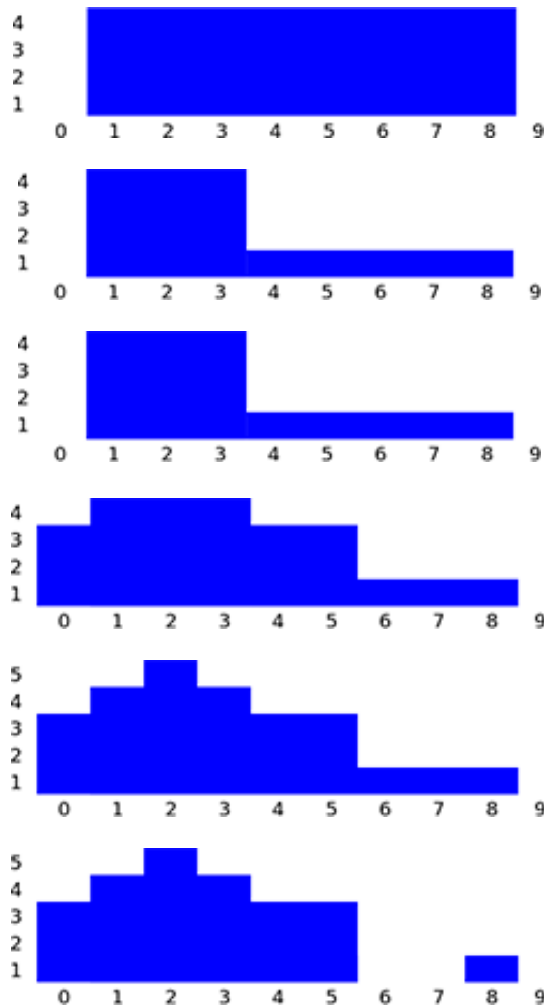
Таны даалгавар бол багананы сүүлчийн хэлбэрийг тогтоох юм.

Жишээ

Бид 10 баганатай ханыг 6-н үйлдлээр барих байсан гэж саная. Дараах хүснэгтэд байгаа бүх завсруудын захын утгууд оролцоно. Доор үйлдэл болгоны дараах диаграммыг харуулав.

үйлдэл	төрөл	завсар	өндөр
0	нэмэх	1-с 8-р багана	4
1	хасах	4-с 9-р багана	1
2	хасах	3-с 6-р багана	5
3	нэмэх	0-с 5-р багана	3
4	нэмэх	2-р багана	5
5	хасах	6-с 7-р багана	0

Эхлээд бүх багананууд нь хоосон байсан тул 0-р үйлдлийн дараа 1-с 8-р багананууд бүгд 4-н тоосготой болсон. 0 ба 9-р багананууд нь хоосон хэвээр. 1-р үйлдлээр, 4-с 8-р багананууд бүгд 1 тоосготой болтол нь тоосгуудыг хасна, 9-р багана хоосон хэвээр. Завсарт ороогүй 4-с бага багананууд өөрчлөгдөөгүй үлдсэн. 2-р үйлдлээр 3-с 6-р баганануудад 5-с их тоосготой багана байхгүй тул өөрчлөлт гарахгүй. 3-р үйлдлийн дараа 0, 4, 5-р баганануудын тоосгоны тоо 3 болж нэмэгдэнэ. 4-р үйлдлээр 2-р багана 5 тоосготой болно. 5-р үйлдлээр 6, 7-р багананы тоосгууд байхгүй болно.



Даалгавар

k үйлдлүүдийн мэдээлэл өгөгдөхөд бүх үйлдэл дууссаны дараа багана бүрд байгаа тоосгоны тоог олно уу. Та `buildWall` гэсэн функцыг хэрэгжүүлэх ёстой.

- `buildWall(n, k, op, left, right, height, finalHeight)`
 - n : Ханан дахь баганы тоо.
 - k : үйлдлүүдийн тоо.
 - op : k хэмжээтэй хүснэгт; $op[i]$ нь бүх $0 \leq i \leq k - 1$ байх i -р үйлдлийн хувьд: 1 байвал *нэмэх* үйлдэл, 2 байвал *хасах* үйлдэл.
 - $left$ болон $right$: k хэмжээтэй хүснэгтүүд; бүх $0 \leq i \leq k - 1$ байх i -р үйлдлийн завсар нь $left[i]$ баганаас эхлэх ба $right[i]$ баганаар дуусна ($left[i]$ болон $right[i]$ төгсгөлүүд нь завсаргаа орно). Танд үргэлж $left[i] \leq right[i]$ байхаар өгөгдөнө.
 - $height$: k хэмжээтэй хүснэгт; $height[i]$ нь бүх $0 \leq i \leq k - 1$ байх i -р үйлдлийн өндөр.
 - $finalHeight$: n хэмжээтэй хүснэгт; та i -р баганд байх сүүлийн тоосгоны тоог $finalHeight[i]$ -д хадгална.

Дэд даалгаврууд

Бүх дэд даалгавруудад бүх үйлдлүүдийн өндрийн параметр нь сөрөг биш, 100,000-с хэтрэхгүй бүхэл тоо байна.

дэд даалгавар	оноо	n	k	тэмдэглэл
1	8	$1 \leq n \leq 10,000$	$1 \leq k \leq 5,000$	ямар ч нэмэлт хязгаарлалтгүй
2	24	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	бүх нэмэх үйлдлүүд нь бүх хасах үйлдлүүдийн өмнө байна
3	29	$1 \leq n \leq 100,000$	$1 \leq k \leq 500,000$	ямар ч нэмэлт хязгаарлалтгүй
4	39	$1 \leq n \leq 2,000,000$	$1 \leq k \leq 500,000$	ямар ч нэмэлт хязгаарлалтгүй

Хэрэгжүүлэх тайлбар

Та wall.c, wall.cpp эсвэл wall.pas гэж нэрлэгдэх яг ганц л файл илгээнэ. Уг файл дээр дүрслэгдсэн дэд програмыг доорх хэвийн дагуу хэрэгжүүлнэ. C/C++ програмын хувьд та wall.h тойлгой файлыг агуулах хэрэгтэй.

C/C++ програм

```
void buildWall(int n, int k, int op[], int left[], int right[], int height[], int finalHeight[]);
```

Паскал програм

```
procedure buildWall(n, k : longint; op, left, right, height : array of longint; var finalHeight : array of longint);
```

Жишээ шалгагч (grader)

Жишээ шалгагч нь оролтын доорх форматаар уншина:

- мөр 1: n, k .
- мөр $2 + i$ ($0 \leq i \leq k - 1$): $op[i], left[i], right[i], height[i]$.