

هیپ گم شده

- محدودیت زمان: ۱.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آرایه A تشکیل یک مکس هیپ می‌دهد، یعنی برای هر i ($2 \leq i \leq n$) داریم که $A_{\lfloor i/2 \rfloor} > A_i$ اما متأسفانه برخی از خانه‌های آرایه گم شده و درون آنها -1 نوشته شده است. آیا می‌توانید آرایه را جوری پر کنید (جای مقادیر -1 مقدار جدیدی قرار دهید) که آرایه باز هم تشکیل یک مکس هیپ بدهد؟

ورودی

ورودی شامل دو خط است که در خط اول عدد طبیعی n که اندازه آرایه است آمده است.

$$1 \leq n \leq 10^5$$

خط دوم شامل آرایه، که شامل اعداد بین 1 تا 10^9 و همچنین -1 است، که با فاصله از هم آمده‌اند.

خروجی

اگر نمی‌توان مقادیر -1 را جوری جایگزین کرد که آرایه تشکیل مکس هیپ دهد در خروجی -1 چاپ کنید. وگرنه آرایه بازبایی شده را در یک خط با فاصله از هم چاپ کنید. دقت کنید که اعداد آرایه خروجی باید بین 1 تا 10^9 باشند.

مثال

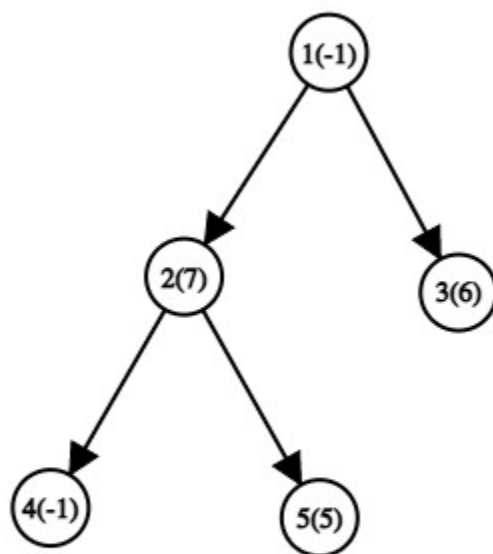
ورودی نمونه ۱

```
5
-1 7 6 -1 5
```

خروجی نمونه ۱

8 7 6 1 5

شکل هیپ ورودی:



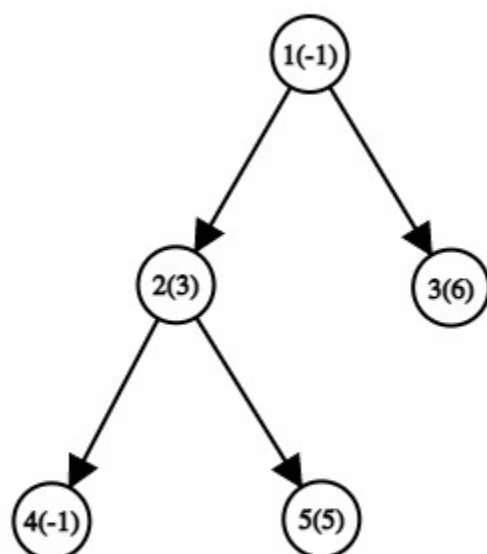
ورودی نمونه ۲

5
-1 3 6 -1 5

خروجی نمونه ۲

-1

شکل هیپ ورودی:



امتیازی: پارتیه‌مونی

- محدودیت زمان: ۲.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک مهمانی که به مدت m دقیقه و دارای n مهمان است در حال برگزاری است. مهماندار می‌داند مهمان i ام در دقیقه l_i وارد مهمانی شده و دقیقه‌ی r_i مهمانی را ترک می‌کند، همچنین می‌داند مهمان i ام در جیب خود c_i دلار پول دارد. وی می‌خواهد بداند که در هر دقیقه‌ی مهمانی (دقایق 1 تا m) پولدارترین مهمان حاضر در آن زمان چند دلار دارد، آیا می‌توانید اطلاعات خواسته‌ی مهماندار را به او بدهید؟

ورودی

خط اول ورودی شامل دو عدد n و m می‌شود که با فاصله از هم جدا شده‌اند.

در n خط بعدی برای هر مهمان اعداد l_i ، r_i و c_i با فاصله از هم آمده‌اند.

$$1 \leq n \leq 150000$$

$$1 \leq m \leq 300000$$

$$1 \leq l_i \leq r_i \leq m$$

$$1 \leq c_i \leq 10^9$$

نکته: تضمین می‌شود تمام c_i ها متمایز هستند.

خروجی

در تنها خط خروجی بیشترین مقدار دلار موجود در جیب مهمانان حاضر در دقیقه i ام را برای تمام دقایق 1 تا m با

فاصله از هم به ترتیب از چپ به راست بنویسید. اگر در یک زمان مهمانی حاضر نباشد بیشترین مقدار 0 است.

مثال

ورودی نمونه ۱

```
4 10
5 6 2
8 10 9
3 5 10
9 10 6
```

خروجی نمونه ۱

```
0 0 10 10 10 2 0 9 9 9
```

توضیح

ابتدا در دقیقه 3 مهمان شماره سه وارد می شود در نتیجه بیشترین مقدار پول 10 خواهد شد (خروجی را از دقیقه 1 شروع کنید) و تا دقیقه 5 بیشترین مقدار پول را دارد سپس مهمانی را ترک میکند و مهمان شماره یک که در دقیقه 5 وارده شده پولدارترین فرد هست و بعد از ترک مهمانی هیچ فردی وجود ندارد در نتیجه 0 باید چاپ شود سپس در دقیقه 8 مهمان شماره دو وارد میشود و باید 9 را چاپ کنید.

ورودی نمونه ۲

```
4 10
2 10 10
5 6 5
5 9 9
3 6 8
```

خروجی نمونه ۲

0 10 10 10 10 10 10 10 10 10 10

اولین تشریحی

الف) اعداد زیر را به ترتیب از چپ به راست وارد یک درخت قرمز-سیاه خالی کنید و در هر insertion ، در صورت نقض یکی از خصوصیات درخت قرمز-سیاه، case آن را بیان کنید و مراحل تصحیح آن را نشان دهید.

16, 33, 21, 17, 13, 25, 12, 19, 26, 40, 22, 24

ب) اعداد زیر را از درخت به دست آمده حذف کنید، مراحل حذف را توضیح دهید و در صورت نقض هر یک از خصوصیات درخت قرمز-سیاه، case آن را بیان کنید و مراحل تصحیح آن را نشان دهید.

21, 24, 16

ج) ارتفاع اصلی و ارتفاع سیاه درخت نهایی را به دست آورید.

دومین تشریحی

الف) بررسی کنید که آرایه زیر max-heap هست یا خیر. اگر نیست BUILD-MAX-HEAP را روی آن اجرا کنید و مراحل اجرای آن را نشان دهید.

21, 15, 17, 25, 23, 30, 24, 28, 13, 12, 20, 19, 10

ب) به ازای فراخوانی های زیر، درخت نهایی را رسم کنید. (فراخوانی ها را به ترتیب قرار گرفته شده انجام دهید).

۱. HEAP-EXTRACT-MAX (A)

۲. HEAP-INCREASE-KEY (A , i=4 , key=40) (منظور از i ، خانه ای از آرایه که باید به مقدار k افزایش یابد)

۳. MAX-HEAP-INSERT (A , key=38) (منظور از k مقداری است که باید وارد هیپ شود)

سومین تشریحی

بوسیله یک تغییر در یک درخت قرمز-سیاه ساختمان داده‌ای بسازید که اعمال زیر را با این هزینه‌ها انجام دهد:

| تایع | هزینه |
|-----------|--------------|
| Insert(x) | $O(\log(n))$ |
| Delete(x) | $O(\log(n))$ |
| Find(x) | $O(\log(n))$ |
| Count(x) | $O(\log(n))$ |

تابع Find به عنوان ورودی یک عدد می‌گیرد و در صورت وجود این عدد یک اشاره گر به گره‌ای که این مقدار را دارد برمی‌گرداند. تابع Count نیز تعداد گره‌های کوچکتر از x را نشان می‌دهد.