

## ابرفصف

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

برای یک ابر صف سه نوع درخواست تعریف می‌شود:

- $+ d t$  : به تعداد  $t$  عدد  $d$  به ته صف اضافه شود.
  - $- t$  : به تعداد  $t$  عدد از سر صف پاک شود.
  - $?$  : چه عددی سر صف قرار دارد؟ (در صورت خالی بودن صف جواب درخواست empty است).
- برنامه‌ای بنویسید که  $q$  درخواست از ورودی بخواند و به ازای هر درخواست از نوع  $?$  مقدار سر صف را چاپ کند.

## ورودی

خط اول ورودی شامل  $q$  تعداد درخواست‌هاست و در  $q$  خط بعدی در هر خط یک درخواست از سه نوع تعریف شده آمده است.

$$1 \leq q \leq 10^5$$

- $+ d t$  : در درخواست‌های از نوع اضافه کردن محدودیت زیر برقرار است.

$$1 \leq t, d \leq 10^9$$

- $- t$  : در درخواست‌های از نوع پاک کردن محدودیت زیر برقرار است.

$$0 \leq t \leq 10^9$$

## خروجی

به ازای هر درخواست از نوع  $?$  مقدار عدد سر صف (یا در صورت خالی بودن صف empty) را در یک خط جدید

بنویسید.

## مثال

### ورودی نمونه ۱

7  
+ 4 3  
+ 3 2  
?  
+ 1 3  
- 4  
+ 1 1  
?

### خروجی نمونه ۱

4  
3

در ابتدا ۳ تا 4 وارد صف می شود سپس ۲ تا 3. بنابراین در خط سوم در فراخوانی ؟ باید عدد 4 که اول صف هست چاپ شود. سپس ۳ تا 1 وارد میشود و ۴ عنصر اول را از ابتدا حذف می کنیم. سپس یک 1 به صف اضافه می کنیم. در آخرین فراخوانی ؟ از آن جایی که ۴ تا عنصر اول صف حذف شده بودند عنصر اول صف 3 است.

### ورودی نمونه ۲

10  
+ 1 3  
+ 4 2  
- 1  
+ 5 3  
+ 2 5  
+ 3 1

$$+ 3 \ 2$$

$$- 6$$

?

?

در ابتدا ۳ تا ۱ و ۲ تا ۴ وارد می شود سپس اولین عنصر صف حذف می شود. سپس به ترتیب ۳ تا ۵ ، ۵ تا ۲ ، یکی ۳ و ۲ تا ۳ وارد می کنیم و به یک همچنین صفی می رسم: [1,1,4,4,5,5,5,2,2,2,2,3,3,3] در آخر ۶ عنصر اول را حذف می کنیم تا به صف زیر برسیم: [5,2,2,2,2,2,3,3,3] در آخر دو فراخوانی ؟ داریم که باید عنصر اول یا ۵ را چاپ کنیم.

## خروجی نمونه ۲

5

5

## راهنمایی:

به جای اینکه اعداد رو دونه دونه به صف اضافه و کم کنیم می توان به صورت بسته ی تعداد و عدد به صف اضافه و کم کنیم. برای مثل:

$$+ 2 \ 3$$

$$+ 1 \ 100$$

$$- 100$$

?

ابتدا به بسته ی ۳ تایی از ۲ ها و بعد به بسته ی ۱۰۰ تایی از ۱ ها به صف اضافه می شود و با برداشتن ۱۰۰ عدد بسته ۳ تایی کامل پاک شده و از بسته ۱۰۰ تایی ۳ تا باقی می ماند.

## درخت گم شده

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۱۲۸ مگابایت

یک درخت جستجوی دودویی با  $n$  راس در نظر بگیرید که هر راس دارای خصوصیات زیر است:

- اندیس: یک عدد منحصر به فرد بین ۱ تا  $n$  که نام راس است.
- مقدار: یک عدد منحصر به فرد بین ۱ تا  $n$  که قوانین جستجوی درختی بر اساس آن اعمال می‌شود. (مقدار یک راس از مقدار تمام رئوس زیر درخت بچه‌ی چپش بزرگ‌تر و از مقدار تمام رئوس زیردرخت بچه‌ی راستش کمتر است).

• اندیس بچه‌ی چپ راس (در صورتی که راس، بچه سمت چپ ندارد مقدار این خصوصیت -1 است).

• اندیس بچه‌ی راست راس (در صورتی که راس، بچه سمت راست ندارد مقدار این خصوصیت -1 است).

ساختار درخت حفظ شده ولی مقدار رئوس گم شده است. برنامه‌ای بنویسید که ساختار درخت را ورودی گرفته و مقدار رئوس را پیدا کرده و چاپ کند.

## ورودی

در خط اول  $n$  تعداد رئوس درخت آمده است و در  $n$  خط بعدی سه عدد  $i$  (اندیس راس)،  $l$  (اندیس راس بچه‌ی چپ) و  $r$  (اندیس راس بچه‌ی راست) با فاصله از هم آمده‌اند.

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

تضمین می‌شود اطلاعات ورودی تشکیل یک درخت دودویی بدهند و ارتفاع درخت حداکثر ۱۰۰ باشد. (دقت کنید که ریشه درخت یکی از گره‌های ورودی است و لزوماً گره اول نیست و باید آن را پیدا کنید)

## خروجی

در تنها خط خروجی مقدار راس اول، مقدار راس دوم، ... و مقدار راس  $n$ ام را با فاصله از هم چاپ کنید.

## مثال

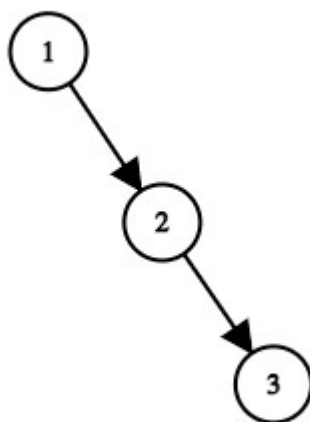
### ورودی نمونه ۱

```
3
1 -1 2
2 -1 3
3 -1 -1
```

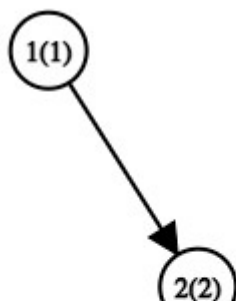
### خروجی نمونه ۱

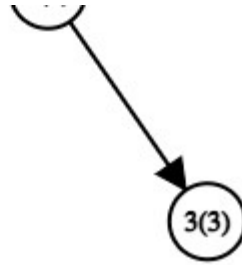
```
1 2 3
```

درخت بدون مقادیر:



درخت همراه با مقادیر:





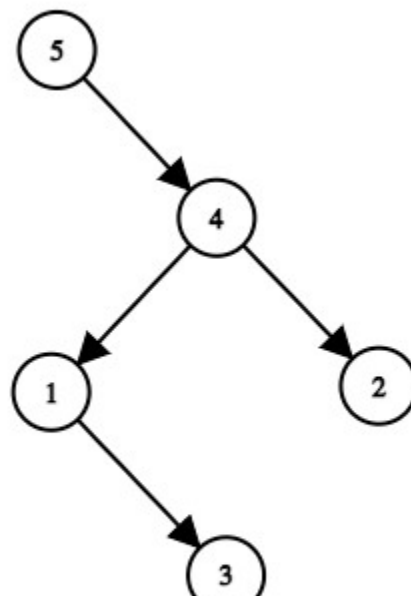
ورودی نمونه ۲

5  
5 -1 4  
1 -1 3  
3 -1 -1  
4 1 2  
2 -1 -1

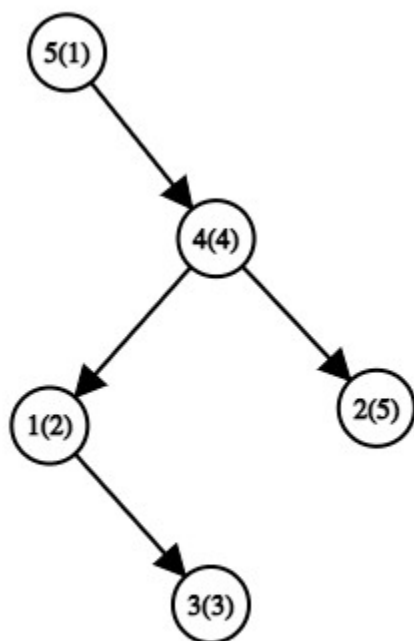
خروجی نمونه ۲

2 5 3 4 1

درخت بدون مقادیر:



درخت همراه با مقادیر:



## راهنمایی

تابع *inOrderTree* را پیاده سازی کنید. ترتیبی که این تابع طی می‌کند همان ترتیبی است که سوال می‌خواهد.

## جعبه‌ها

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

در  $n$  جایگاه  $n$  جعبه به شماره‌های ۱ تا  $n$  قرار داده شده است. در  $m$  حرکت هر دفعه دو عدد  $i$  و  $j$  انتخاب کرده و تمام جعبه‌های موجود در جایگاه  $i$ ام (جایگاه می‌تواند خالی باشد) را برداشته و بر روی جعبه‌های جایگاه  $j$ ام قرار می‌دهیم. برنامه‌ای بنویسید که حرکات را ورودی گرفته و شماره‌ی جعبه‌های جایگاه  $d$ ام را پس از انجام حرکات از پایین به بالا چاپ کند.

## ورودی

در خط اول دو عدد  $n$  (تعداد جعبه‌ها) و  $m$  (تعداد حرکات) با فاصله از هم آمده‌اند. در  $m$  خط بعدی دو عدد  $i$  و  $j$  با فاصله از هم آمده‌اند و در خط آخر عدد  $d$  آمده است.

$$1 \leq n, m \leq 2 \times 10^5$$

$$1 \leq i \neq j \leq n$$

$$1 \leq d \leq n$$

## خروجی

در تنها خط خروجی ابتدا تعداد جعبه‌های موجود در جایگاه  $d$ ام را پس از انجام حرکات چاپ کرده و سپس شماره‌ی جعبه‌های موجود در جایگاه  $d$ ام را از پایین به بالا چاپ کنید.

## مثال

### ورودی نمونه ۱



4 3

1 2

4 3

2 3

3

در ابتدا هر جعبه در جایگاه با شماره همان جعبه قرار گرفته است. ابتدا جعبه 1 بر روی جعبه ی جایگاه شماره 2 یعنی جعبه 2 قرار می گیرد. سپس جعبه 4 بر روی جعبه 3 قرار می گیرد. در گام بعدی جعبه های جایگاه 2 (1و2) را بر روی جعبه های جایگاه 3 (3و4) قرار می دهیم.

### خروجی نمونه ۱

4 3 4 2 1

### ورودی نمونه ۲

4 3

1 4

4 3

3 2

2

در ابتدا جعبه 1 را بر روی 4 قرار می دهیم. سپس جعبه های جایگاه 4 (1و4) را بر روی جعبه 3 قرار می دهیم. سپس جعبه های جایگاه شماره 3 را بر روی جعبه های جایگاه 2 قرار می دهیم.

### خروجی نمونه ۲

4 2 3 4 1

## راهنمایی

برای هر جایگاه یک لینکد لیست بسازید. هر عملیات جابجایی می تواند با استفاده از لینکد لیست در مرتبه زمانی ۱

انجام شود. (با استفاده از عوض کردن `head` و `tail` لیست‌ها) دقت کنید که لینکد لیست را خودتان پیاده سازی کنید چون در زبان‌ها تابعی مانند `addAll` در مرتبه زمانی تعداد اعضای لیست کار می‌کند نه ۱ و همچنین حواستون به جابجایی از یا روی لیست خالی باشد.

## اولین تشریحی

یک صف (*Queue*) را با استفاده از دو پشته (*Stack*) شبیه سازی کنید.

## دومین تشریحی

الگوریتم *Inorder\_Tree\_walk* را به صورت یک الگوریتم غیر بازگشتی بیان کنید.

## امتیازی

فرض کنید دو تا لینکد لیست ساده داریم که طول اولی  $m$  و دومی  $n$  باشد و دقیقاً نمیدانیم که  $n < m$  است یا برعکس. این دو تا لینکد لیست از یک نود خاص به بعد کاملاً دارای نود های مشترک می شوند.

از شما می‌خواهیم الگوریتمی ارائه دهید که در اردر زمانی  $O(m + n)$  این نود خاص که اولین اشتراک این دو لینکد لیست هست را پیدا کند.