

## پیاده سازی داده ساختار

- محدودیت زمان: ۴ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این قسمت باید به pdf تمرین و قالب داده شده مراجعه کرده و توابع مربوط به کلاس ها را که خالی هستند با کد مناسب کامل کنید.

### ورودی

با توجه به قالب داده شده ابتدا یک یا چند آبجکت ایجاد می شود. سپس توابع مشخص شده برای هر کدام صدا زده می شوند که همگی در قالب آمده است و توضیح مربوط به هر کدام در pdf تمرین آمده است.

ترتیب تست ها بدین شکل است :

1- 1 تا 5 : Min Heap

2- 6 تا 15 : Red Black Tree

3- 16 تا 25 : Huffman Tree

### مثال

#### ورودی نمونه ۱

```
make min_heap m1
call m1.heapify(10,5,30,50)
call m1.find_min_child(0)
call m1.heap_pop()
call m1.heap_pop()
call m1.heap_pop()
call m1.heap_pop()
```

```
call m1.find_min_child(-1)
call m1.find_min_child(1)
call m1.find_min_child('salap')
```

خروجی نمونه ۱

```
1
5
10
30
50
out of range index
out of range index
invalid index
```

ورودی نمونه ۲

```
make red_black_tree rb1
call rb1.insert(100)
call rb1.insert(40)
call rb1.insert(300)
call rb1.insert(20)
call rb1.insert(75)
call rb1.insert(57)
call rb1.find_node_color(40)
call rb1.find_node_color(300)
call rb1.find_node_color(20)
call rb1.find_node_color(57)
call rb1.find_node_color(100)
call rb1.find_node_color(75)
```

خروجی نمونه ۲

```
RED
BLACK
```

BLACK  
RED  
BLACK  
BLACK

### ورودی نمونه ۳

```
make huffman_tree h1
call h1.set_letters('a','b','c','d','e','f')
call h1.set_repetitions(1,3,12,13,16,1000)
call h1.build_huffman_tree()
call h1.get_huffman_code_cost()
make huffman_tree h2
call h2.text_encoding('chahi-migholam-garm-sham-va-sard-va-tondkhoo-nabasham')
call h2.get_huffman_code_cost()
```

### خروجی نمونه ۳

1139  
198

## k امین

- محدودیت زمان: ۴ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آلایا به تازگی یاد گرفته است که می تواند با مین هیپ همواره مقدار مینیمم آرایه ای که همواره در حال تغییر است (حذف یک عنصر و اضافه کردن یک عنصر) را بدست آورد. او که دوست دارد خود را به چالش بکشد اکنون میخواهد در آرایه ای که همواره در حال تغییر است  $k$  امین کوچکترین عنصر آرایه را بدست آورد. آیا می توانید به او کمک کنید؟

## ورودی

در خط اول سه عدد  $n$  و  $q$  و  $k$  داده می شود. در خط دوم  $n$  عدد ورودی داده می شود که حالت اولیه ی آرایه است. در  $q$  بعدی هر خط یک عملیات رو آرایه داده می شود.

$$1 \leq n, q, k \leq 10^5$$

$$1 \leq a_i, x \leq 10^9$$

عملیات:

عدد  $x$  با مقدار  $x$  به آرایه اضافه می شود:  $+$

عنصر مینیمم  $k$  ام در صورت وجود از آرایه حذف می شود:  $-$

عنصر  $k$  ام در آرایه سورت شده خروجی داده می شود: `print`

## خروجی

پس از هر `print` خروجی مورد نظر را چاپ کنید. در صورتی که تعداد عناصر آرایه کمتر از  $k$  بود  $-1$  چاپ کنید.

## مثال

### ورودی نمونه ۱

```
6 7 3
1 3 4 7 5 8
print
+ 3
print
-
print
-
print
```

### خروجی نمونه ۱

```
4
3
4
5
```

در ابتدا سومین کوچکترین عنصر برابر ۴ است چون مرتب شده ی آرایه به صورت 1,3,4,5,6,7 است که سومین آن 4 است پس از اضافه شدن عدد 3 آرایه تبدیل به 1,3,3,4,5,6,7 می شود که سومین عدد آن 3 است. پس از حذف عدد 3 آرایه تبدیل به 1,3,4,5,6,7 می شود و بار دیگر سومین عدد کوچک ۴ می شود و پس از حذف عدد 4 آرایه تبدیل به 1,3,5,6,7 تبدیل می شود که سومین عنصر آن 5 است.

### ورودی نمونه ۲

```
3 8 4
1 2 3
print
+ 4
print
+ 10
```

```
+ 20  
print  
-  
print
```

## خروجی نمونه ۲

```
-1  
4  
4  
10
```

در ابتدا تعداد عناصر آرایه برابر ۳ است که از ۴ کمتر است بنابراین خروجی -۱ است. در print بعدی آرایه برابر 1,2,3,4 است بنابراین چهارمین کوچکترین عنصر برابر ۴ است. در print بعدی دو عدد که از 4 بزرگترند به آرایه اضافه شده اند بنابراین خروجی تغییر نمی کند. در نهایت در آخرین print آرایه برابر 1,2,3,10,20 می شود که جواب برابر ۱۰ است.

## بلوک های چسبیده

- محدودیت زمان: ۴ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

آلانا مشغول به رنگ آمیزی یک جدول است. در ابتدا جدول کاملاً سفید است. او در مرحله خانه  $(i, j)$  را که داخل جدول است تصمیم می گیرد به رنگ سیاه در بیاورد. او دوست دارد در هر مرحله تعداد بلوک های متوالی سطری و ستونی ماکسیمال در این جدول را بشمارد. بلوک سطری و ستونی بلوک های به فرم  $1 * l$  و  $l * 1$  هستند که تماماً سیاه هستند و ماکسیمال باشند به این معنی که بلوکی سطری یا ستونی نباشد که طول آن  $l + 1$  باشد و همه ی این  $l$  خانه را داشته باشد. دقت کنید بلوک ماکسیمال  $1 * 1$  می تواند هم به صورت سطری شمرده شود هم ستونی.

## ورودی

در خط اول سه عدد  $n, m, q$  داده می شود که به معنی این است که جدول  $n * m$  است و  $q$  به معنی تعداد رنگ آمیزی هایی است که روی جدول انجام شده است. در هر کدام از  $q$  خط بعدی دو عدد  $i$  و  $j$  داده می شود که به معنی این است خانه  $(i, j)$  در جدول به رنگ سیاه در می آید.

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

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

$$1 \leq i \leq n$$

$$1 \leq j \leq m$$

## خروجی

در  $q$  خط بعدی در هر خط تعداد مستطیل های ماکسیمال را خروجی دهید.

## مثال

### ورودی نمونه ۱

3 3 5  
1 1  
1 2  
2 1  
2 2  
1 3

### خروجی نمونه ۱

2  
3  
4  
4  
5



## آشوب در ریچر

- محدودیت زمان: ۴ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

اندرو رایان که از بیگ ددی ها برای محافظت از لیتل سیسترز قطع امید کرده است ، خود دست به کار شده تا از آنها نگهداری کند. بدین منظور تعدادی از لیتل سیسترها را از چنگ بیگ ددی های سمج با گفتن would you kindly درمی آورد.

حال می خواهد که از آنها در خانه خود که در شهر زیرآبی ریچر قرار دارد محافظت کند. ولی راه و رسم نگهداری از آنها را بلد نیست و از شما می خواهد که او را در این مسیر کمک کنید. او می داند که لیتل سیسترها به نوعی ماده دریایی به نام Adam اعتیاد دارند. بنابراین آنها را در یک ردیف در پشت گیت مخزن Adam قرار می دهد. Adam ها برای تهیه شدن نیازمند زمان هستند. به طوری که در بازه های  $t$  تایی تولید می شوند (اندرو رایان فرانک فانتین را مجبور کرده که هر وقت لیتل سیستری در صف بود شروع به تولید کند).

همچنین در مورد لیتل سیسترها دو مورد را می داند. یکی آنکه آنان تنبل بوده و از لحظه خاصی بلند می شوند تا Adam خود را بگیرند و به لیتل سیسترهای بعد خود نگاه می کنند. اگر کسی بلند شده بود صبر می کنند تا وقتی که برگردد و تمام لیتل سیسترهای بعد خودش نشسته باشند سپس پشت مخزن می روند. مورد دوم آنکه در صورت همزمان بلند شدن هر کدام به گیت نزدیک تر باشد زور بیشتری برای گرفتن Adam دارد چون پلاسمید کمتری از او استخراج شده است! این را هم توجه کنیم که اگر در ثانیه  $t$  لیتل سیستری بلند شد تا Adam بگیرد ، از  $t+1$  به بعد این کار را می کند.

حال اندرو می خواهد بداند که هر لیتل سیستر در چه زمانی Adam خود را دریافت می کند.



## ورودی

در خط اول  $n$  ، تعداد لیتل سیسترها و  $t$  ، مدت زمانی که طول می‌کشد یک واکسن Adam تولید شود را دریافت می‌کنید.

در خط دوم زمان بلند شدن هر لیتل سیستر را به ترتیب دریافت می‌کنید (به ترتیب نزدیک بودن به گیت مخزن Adam).

## خروجی

در تنها خط برای هر لیتل سیستر زمانی که طول می‌کشد تا Adam دریافت کند را چاپ می‌کنید.

## محدودیت ها

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

$$1 \leq t < 2 * 10^7$$

## مثال

### ورودی نمونه ۱

6 100  
0 95 300 200 1000 0

### خروجی نمونه ۱

100 200 400 300 1100 500

توضیح : برای هر نفر خروجی اینگونه است:

در زمان  $t = 0$  لیتل سیستر اول بلند شده تا Adam دریافت کند و در  $t = 100$  دریافت می‌کند.

در زمان  $t = 95$  لیتل سیستر دوم بلند شده تا Adam دریافت کند و در  $t = 200$  دریافت می‌کند.

در زمان  $t = 300$  لیتل سیستر سوم بلند شده تا Adam دریافت کند و با توجه به اینکه نفر چهارم زودتر از او Adam را می‌گیرد باید صبر کند تا Adam بعدی تولید شود و در  $t = 400$  آدام دریافت می‌کند.

در زمان  $t = 200$  لیتل سیستر چهارم بلند شده تا Adam دریافت کند و در  $t = 300$  دریافت می‌کند.

در زمان  $t = 1000$  لیتل سیستر پنجم بلند شده تا Adam دریافت کند و در  $t = 1100$  دریافت می‌کند.

در زمان  $t = 0$  لیتل سیستر ششم بلند شده تا Adam دریافت کند ولی در زمان‌های 100 200 300 400 آدام به ترتیب توسط نفر اول دوم چهارم سوم گرفته می‌شود (چون به مخزن نزدیک تر هستند) و در  $t = 500$  آدام دریافت می‌کند.