

چیندن در BST

اعداد زیر را به ترتیب از چپ به راست در یک BST بچینید.

2, 7, 8, 10, 1, 5, 3

هنر مرتب سازی هیپ

- محدودیت زمان: ۱ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

برنامه ای بنویسید که آرایه ای از اعداد صحیح را با طول دلخواه از کاربر گرفته و سپس با استفاده از مبحث هیپ آن را بصورت صعودی مرتب کند.

ورودی

ورودی تنها شامل دو خط است که در خط اول تعداد عضو های آرایه توسط کاربر وارد می شود و در خط بعد خود عضو های آرایه وارد می شوند.

خروجی

در یک خط باید آرایه ای که بصورت صعودی مرتب شده است چاپ شود.

مثال

ورودی نمونه ۱

5
15 20 1 16 4

خروجی نمونه ۱

1 4 15 16 20

ورودی نمونه ۲

1
451

خروجی نمونه ۲

451

نمایش درخت

ورودی

در بخش اول در خط اول پیمایش inorder و در خط دوم پیمایش level-order داده می شود .

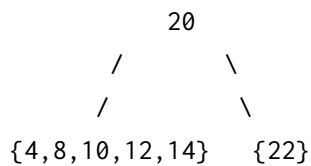
`in[] = {4, 8, 10, 12, 14, 20, 22};`

`level[] = {20, 8, 22, 4, 12, 10, 14};`

خروجی

در خروجی درخت یکتا ساخته شده بر اساس ورودی ها را نمایش دهید . (نمایش بصورت گرافیکی باشد)

خروجی نمونه ۱



تشریحات درخت

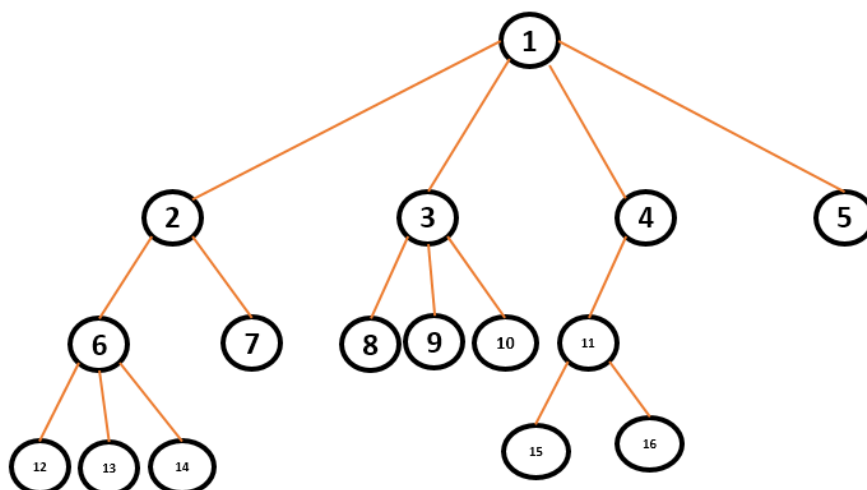
توجه: سوالات تشریحی است و نیازی به کد و حتی سودو کد نیست .

1- درخت generic یا همان n-ary tree زیر را به روش های زیر پیمایش کنید.(نوشتن جواب نهایی کافی است.)

۱. inorder

۲. preorder

۳. postorder



توجه: نوشتن خروجی برای هریک از حالات کافی است.

پیشنهاد: در صورت تمایل درباره پیاده سازی هر یک از موارد بالا و همچنین ساخت یک generic tree با کمک اطلاعاتی که کاربر وارد میکند، مطالعه آزاد داشته باشید.

2- الگوریتم زیر را در نظر بگیرید:

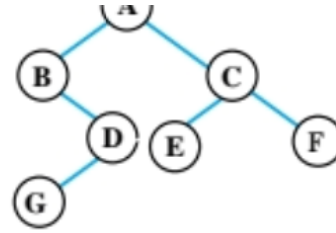
```
void traverse (ptr p){
```



```

stack s; ptr q;
push (p,s);
while (!empty (s)){
    q = pop (s);
    if (q != NULL){ push (q → left,s);
                    print (q → label);
                    push (q → right,s);
                  }
}

```



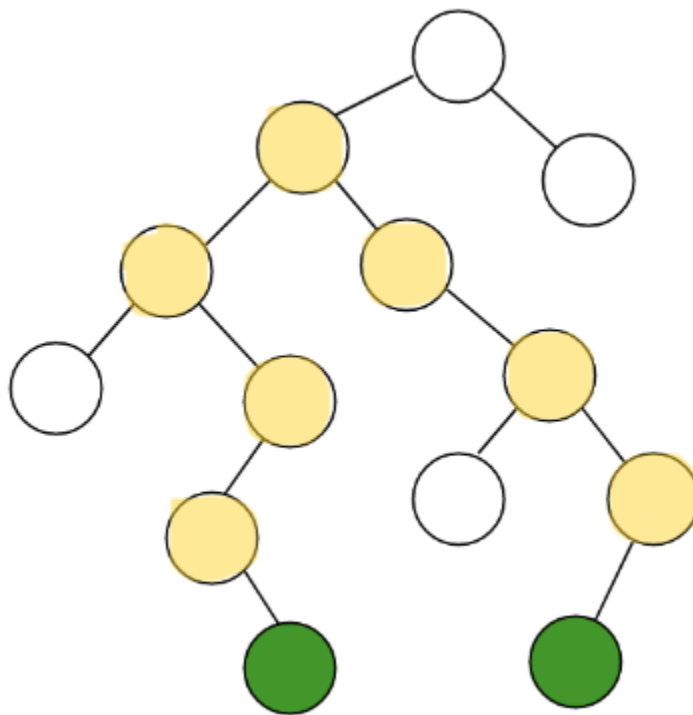
درخت بالا را طبق این الگوریتم پیمایش کنید. (نوشتن جواب نهایی کافی نیست و مراحل را به کمک stack نمایش دهید.)

قطر درخت

- محدودیت زمان: ۱ ثانیه

- محدودیت حافظه: ۲۵۶ مگابایت

قطر یک درخت تعداد گره های موجود در طولانی ترین مسیر بین دو برگ از درخت است.



برای مثال در درخت بالا طولانی ترین مسیر مربوط به دو گره سبز رنگ است. همانطور که مشخص است قطر این درخت 9 است.

توجه کنید که خود گره های سبز رنگ نیز جزء قطر هستند.

حال، برنامه‌ای بنویسید که پیمایش‌های *Inorder* و *Postorder* یک درخت را گرفته و قطر آن را بدست آورد.

ورودی

ورودی شامل سه خط است که در خط اول تعداد گره ها n در خط دوم پیمایش Inorder و در خط سوم پیمایش Postorder درخت آمده است.

خروجی

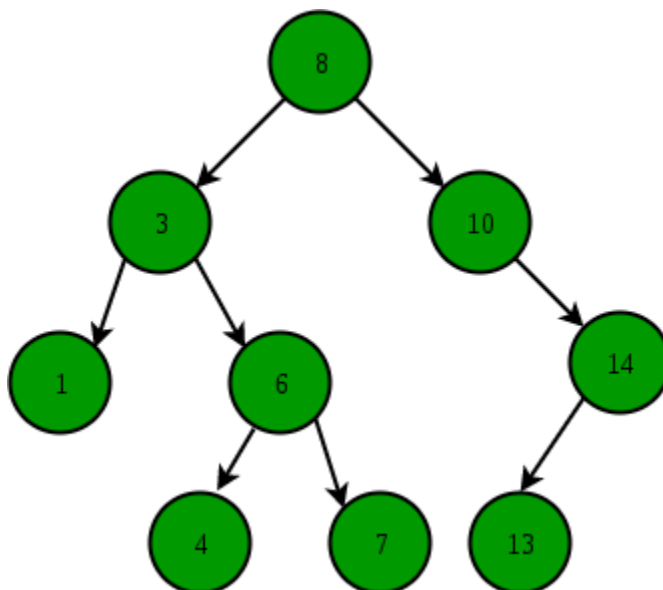
خروجی برنامه یک عدد است که قطر درخت را نشان میدهد.

ورودی نمونه ۱

```
9
1 3 4 6 7 8 10 13 14
1 4 7 6 3 13 14 10 8
```

خروجی نمونه ۱

7



درخت بالا از دو پیمایش داده شده بدست آمده است که با توجه به آن قطر درخت 7 است.

پس‌ترتیب

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

برنامه‌ای بنویسید که با گرفتن «پیش‌ترتیب» (pre-order) و «میان‌ترتیب» (in-order) یک درخت دودویی، پیمایش «پس‌ترتیب» (post-order) آن را به عنوان خروجی بدهد. اعداد درخت دودویی متمایز هستند.

ورودی

در ابتدا عدد n تعداد رئوس درخت می‌آید. در دو خط بعدی، در هر خط n عدد با فاصله از هم می‌آیند که به ترتیب نشان‌دهنده‌ی نمایش «میان‌ترتیب» و «پیش‌ترتیب» درخت خواهند بود.

$$1 \leq n \leq 3 \times 10^5$$

خروجی

تعداد n عدد با فاصله در یک خط بنویسید که نشان‌دهنده‌ی نمایش پس‌ترتیب درخت متناظر باشد.

مثال

ورودی نمونه ۱

```
9
6 2 1 4 3 5 7 9 8
1 2 3 4 5 6 7 8 9
```

خروجی نمونه ۱

1 3 5 4 2 8 9 7 6

زیردرخت

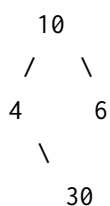
دو درخت باینری به شما داده شده است

برنامه ای بنویسید که بررسی کند آیا یکی از درخت ها زیردرخت دیگری است یا نه

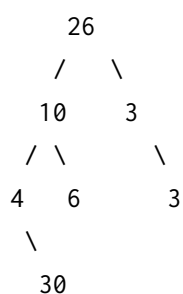
نتیجه را در خروجی چاپ کنید

ورودی

Tree S



Tree T



خروجی

S is subtree of tree T

راهنمایی : برای گرفتن ورودی میتوانید درخت های ورودی سوال را در برنامه hard code کنید و برنامه را برای آن ها اجرا کنید و برنامه باید به ازای ورودی داده شده در صورت سوال پاسخ درست بدهد ورودی را میتوانید به روش دلخواه خود(ارایه, لینک لیست و ...) در کد پیاده کنید

