

تمرین شماره 1  
Time Complexity  
and  
Recursion



ساختمان های داده و الگوریتم - پاییز 1400

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراح تمرین: سینا کمالی

۱۴۰۰/۸/۳۰

استاد: دکتر فیلی

---

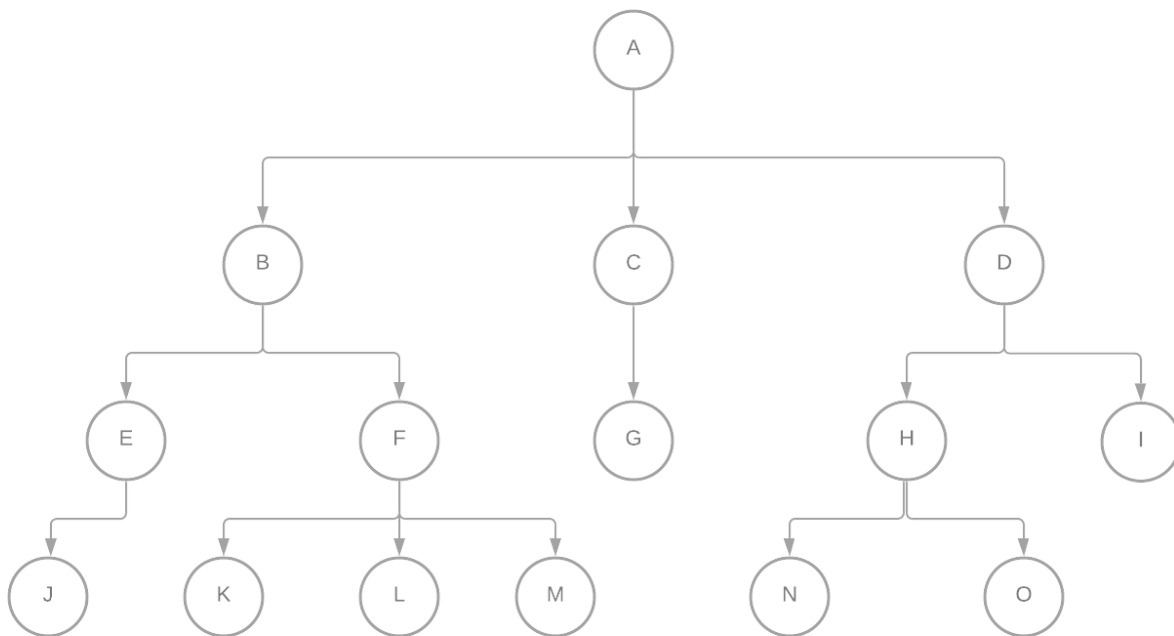
شماره دانشجویی خود را به صورت 81019abcd در نظر بگیرید و سوالات را طبق فرمول داده شده بر اساس شماره دانشجویی خود حل کنید.

(1) پیمایش های خواسته شده را با توجه به باقی مانده  $a+d$  بر 3 یادداشت کنید.

0- Pre Order Traversal

1- Post Order Traversal

2- In Order Traversal



جواب:

1) Preorder: ABEJFKLMCGDHNOI

2) Postorder: JEKLMFBGHCNOHIDA

2) با توجه به باقی مانده  $b+c$  بر 3، از پیمایش postorder داده شده را تبدیل به درخت عبارت متناظر با آن بکنید

و راه حل تولید درخت را به طور کامل بنویسید.

1-  $ab+cd+*ef*G++$

2-  $abc*+de+fg+*/$

3-  $ab+c/defg+-+*$

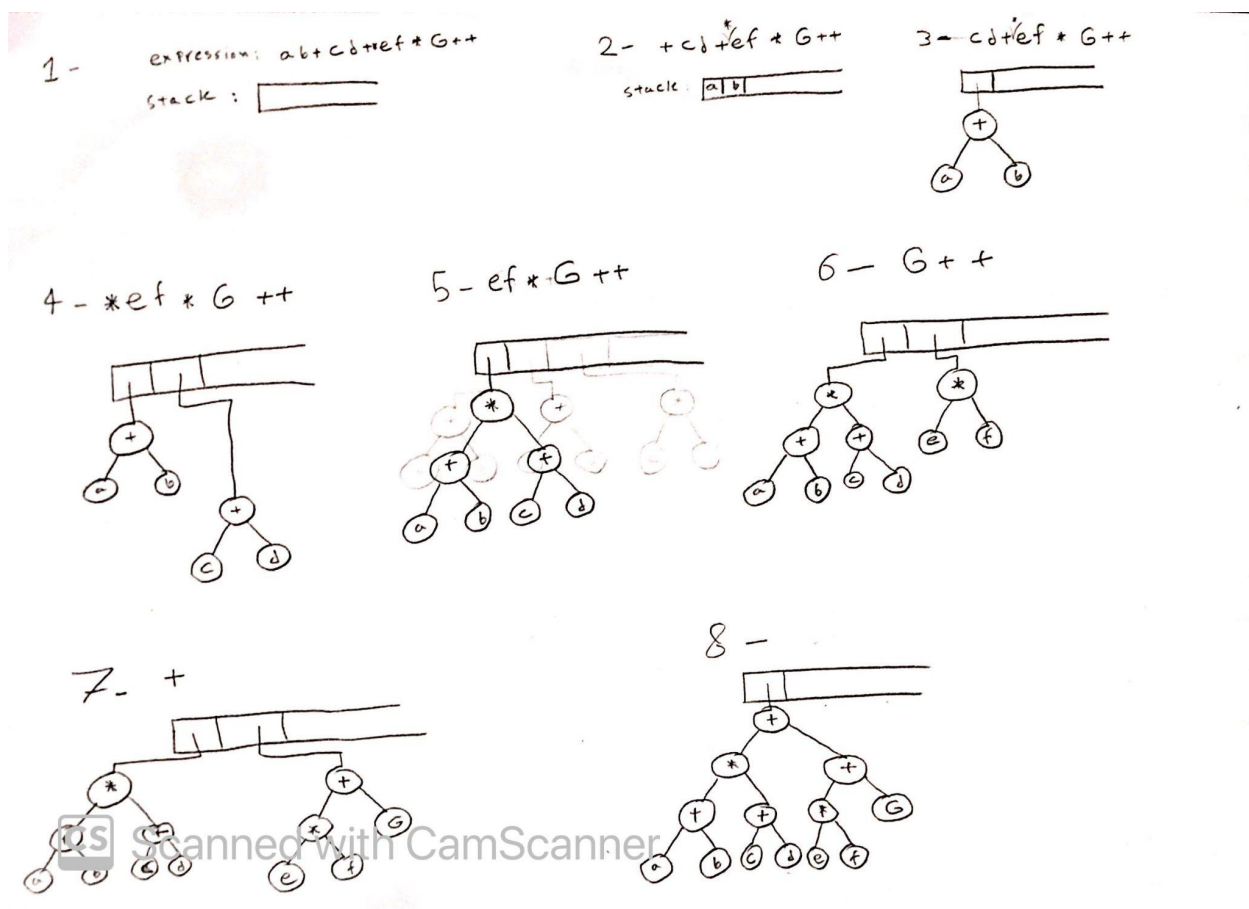
راه حل:

راه تبدیل توصیف postorder یک درخت عبارت به درخت آن، تولید یک stack و push کردن به هنگام رسیدن به

متغیرها و pop کردن دو تا از آنها و push کردن زیر درخت حاصلشان به هنگام رسیدن به یک عملگر است. به

همین صورت عمل میکنیم:

-1



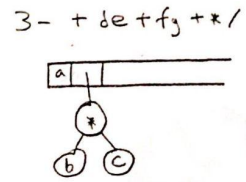
-2

1- expression:  $abc * + de + fg + * /$  2-  $* + de + fg + * /$   
 stack: 

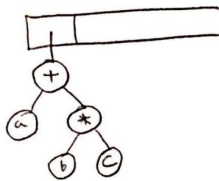
--	--	--

 stack: 

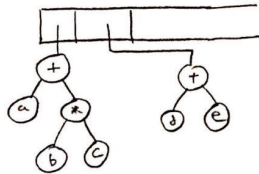
a	b	c
---	---	---



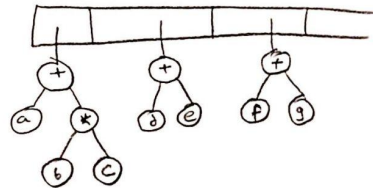
4-  $de + fg + * /$



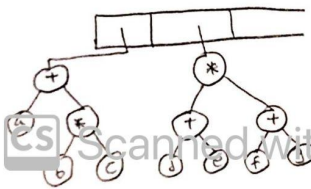
5-  $fg + * /$



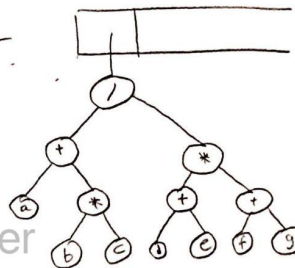
6-  $* /$




7-  $/$

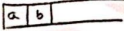


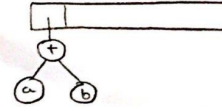
8-



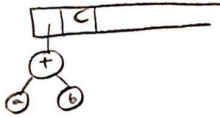
1- expression:  $ab+c/de fg+-+*$  2-  $+c/de fg+-+*$  3-  $c/de fg+-+*$

Stack: 

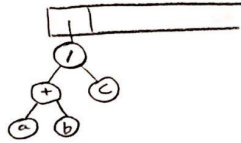
Stack: 



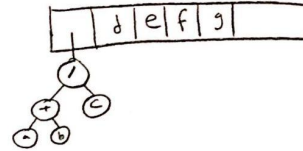
4-  $/de fg+-+*$



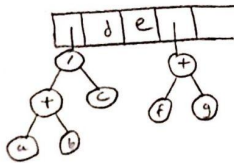
5-  $de fg+-+*$



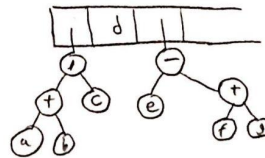
6-  $+-+*$



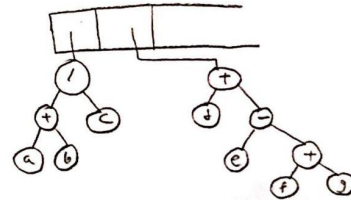
7-  $-+*$



8-  $+*$

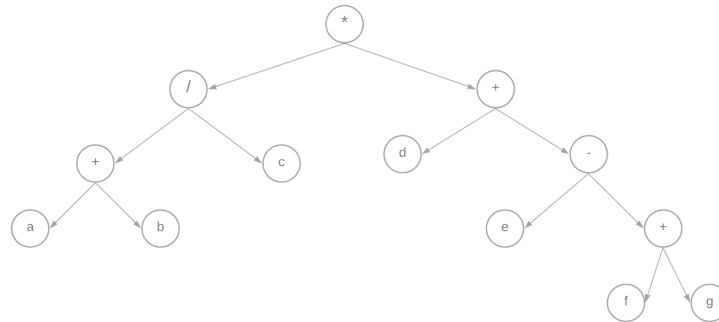


9-  $*$



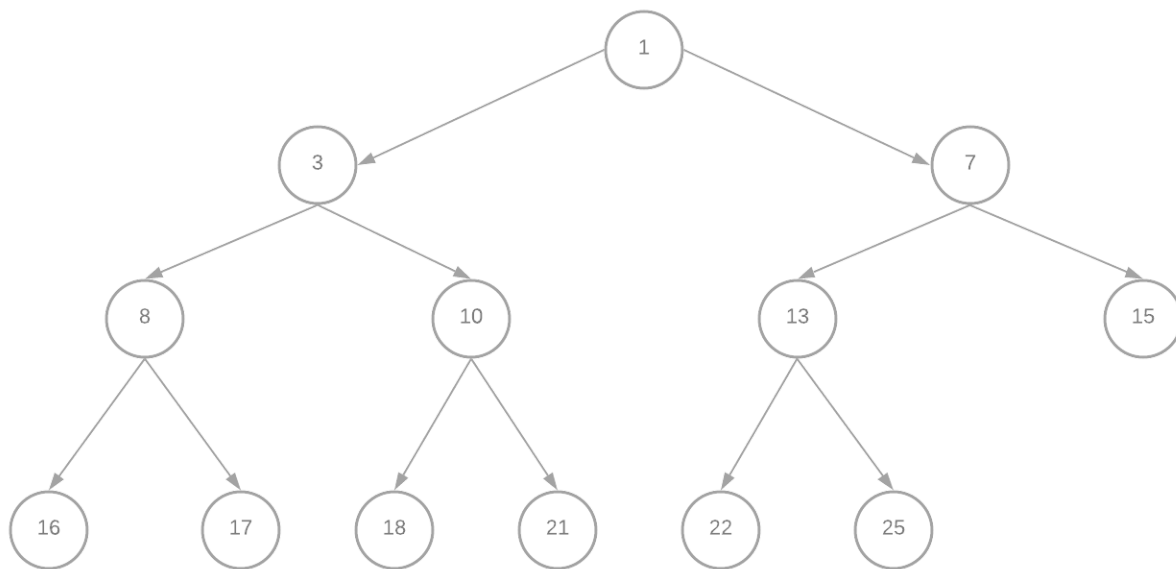
final below

CS Scanned with CamScanner



(3) با توجه به باقی مانده  $b + c + 2a$  بر 3، اعمال داده شده را به ترتیب بر روی درخت min heap زیر انجام

دهید و تمامی مراحل را به طور کامل شرح دهید.



1- insert 6, delete 3

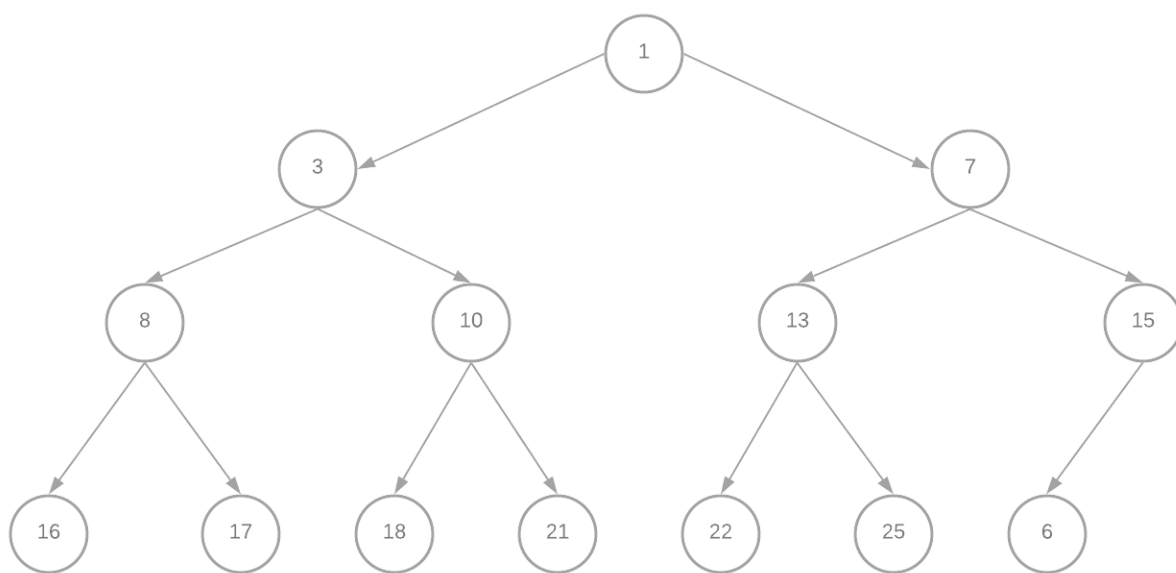
2- delete 1, insert 12

3- delete 10, insert 0

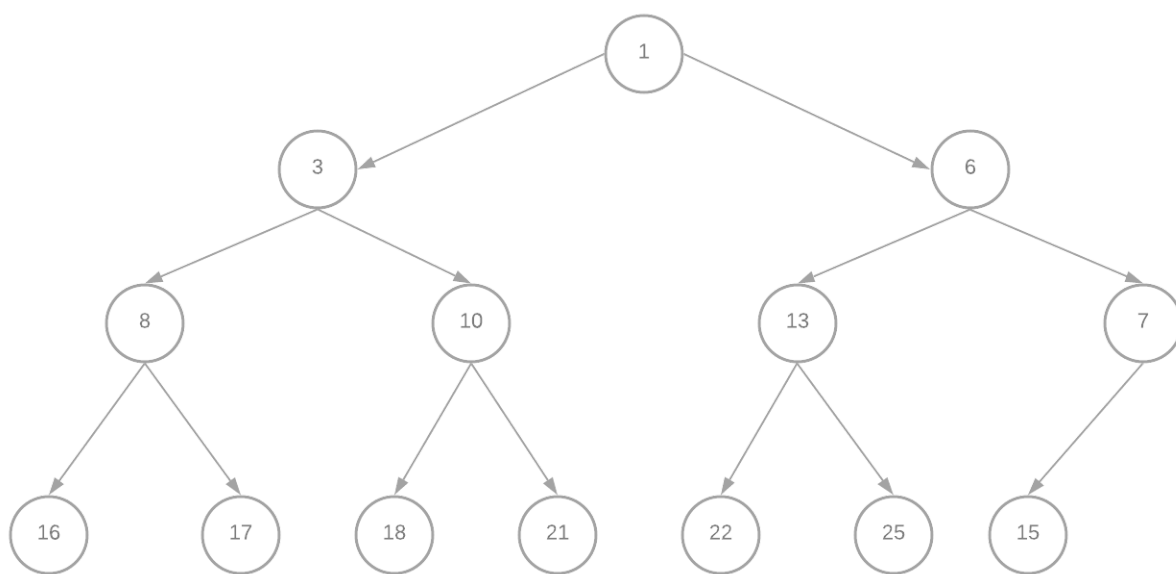
راه حل:

-1

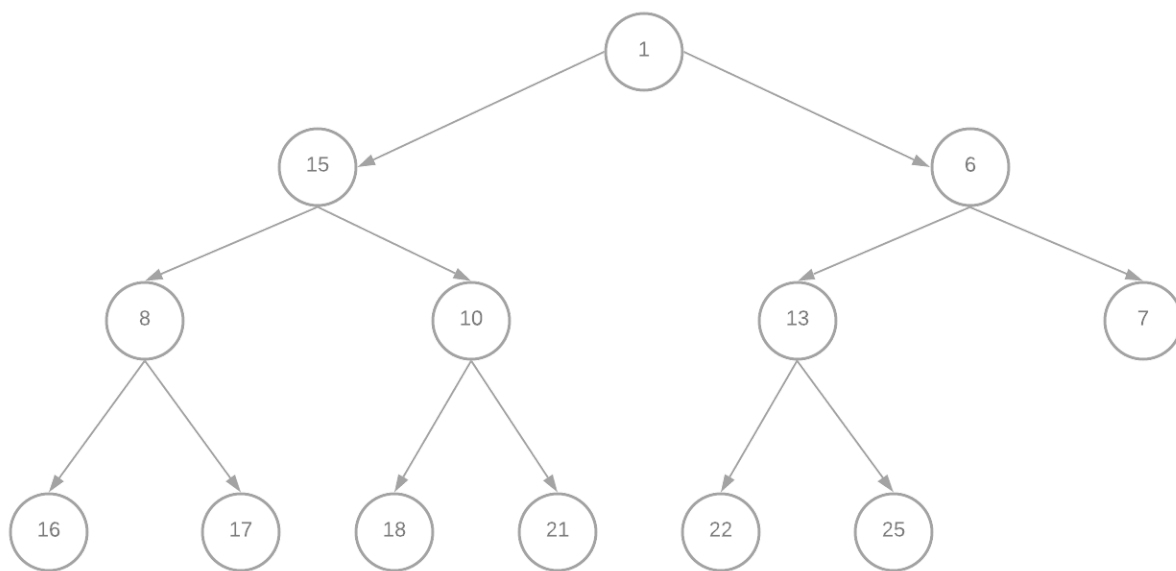
ورود 6: ابتدا 6 را در اولین جایگاه خالی برگ های درخت وارد میکنیم.



سپس مادامی که 6 از پدر خود کوچکتر است جای آن را با پدرش عوض میکنیم. پایان ورود 6.

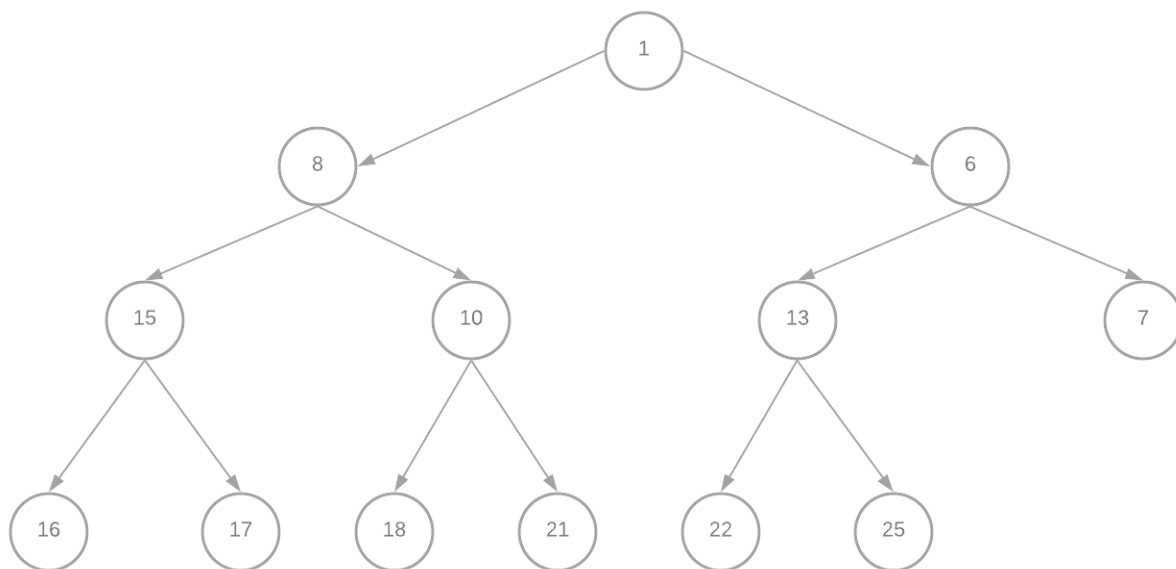


پاک کردن 3: ابتدا 3 را پاک میکنیم و آخرین گره درخت را به جای آن قرار میدهیم. (گره 6)



سپس عمل heapify را انجام می‌دهیم. در bubble down کردن در min heap، دقت کنید که اگر گره‌ای از هر

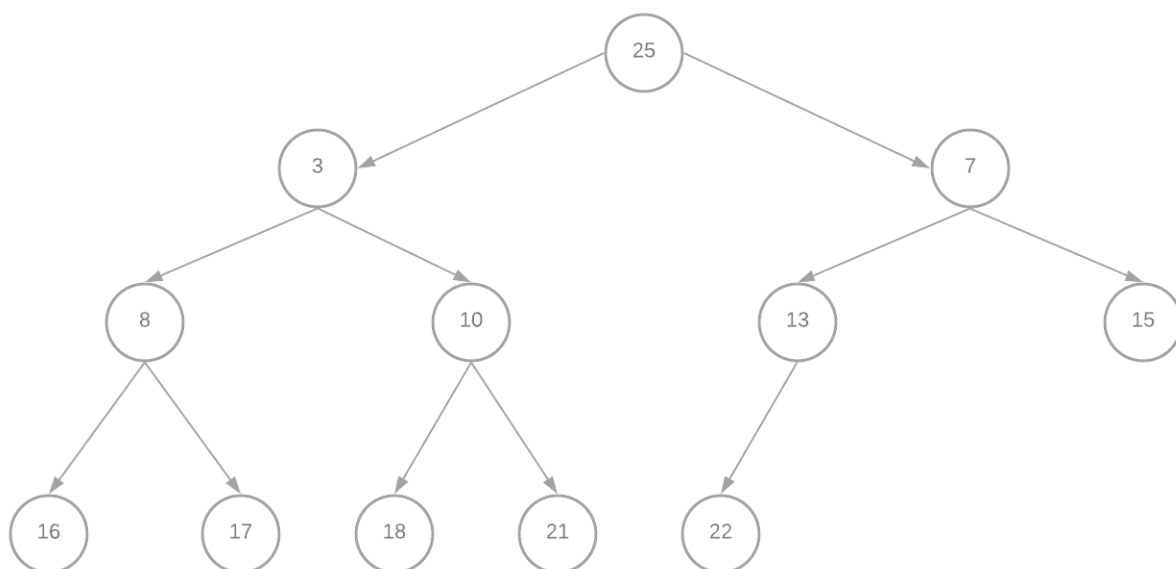
دو فرزند خود کوچکتر بود، آن را با فرزند کوچکتر تعویض کنید.





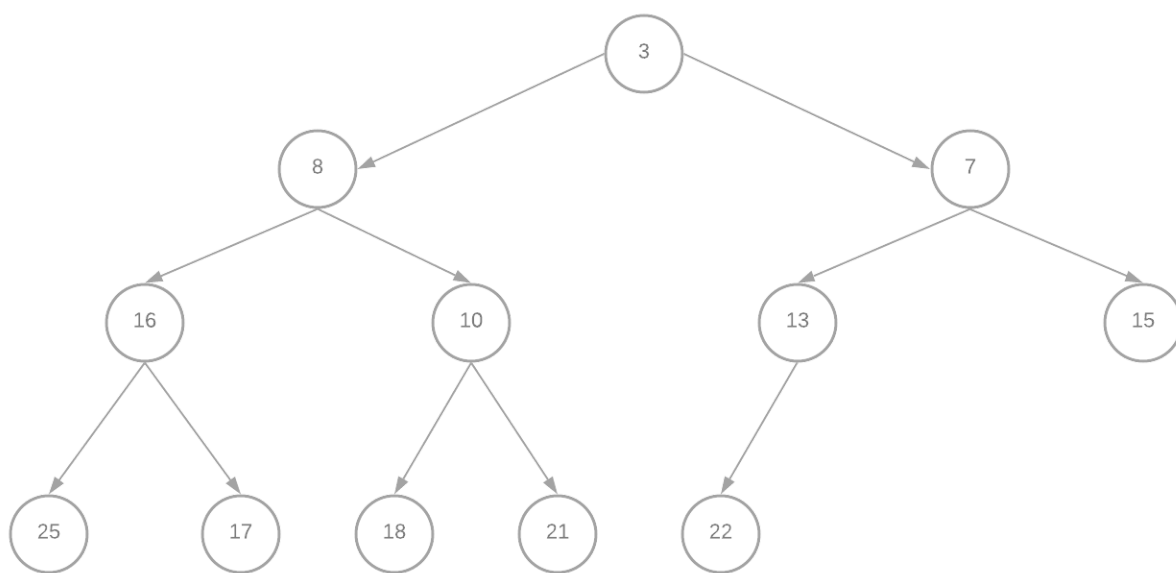
-2

پاک کردن 1: ابتدا 1 را حذف میکنیم و آخرین برگ درخت را به جای آن قرار میدهم.

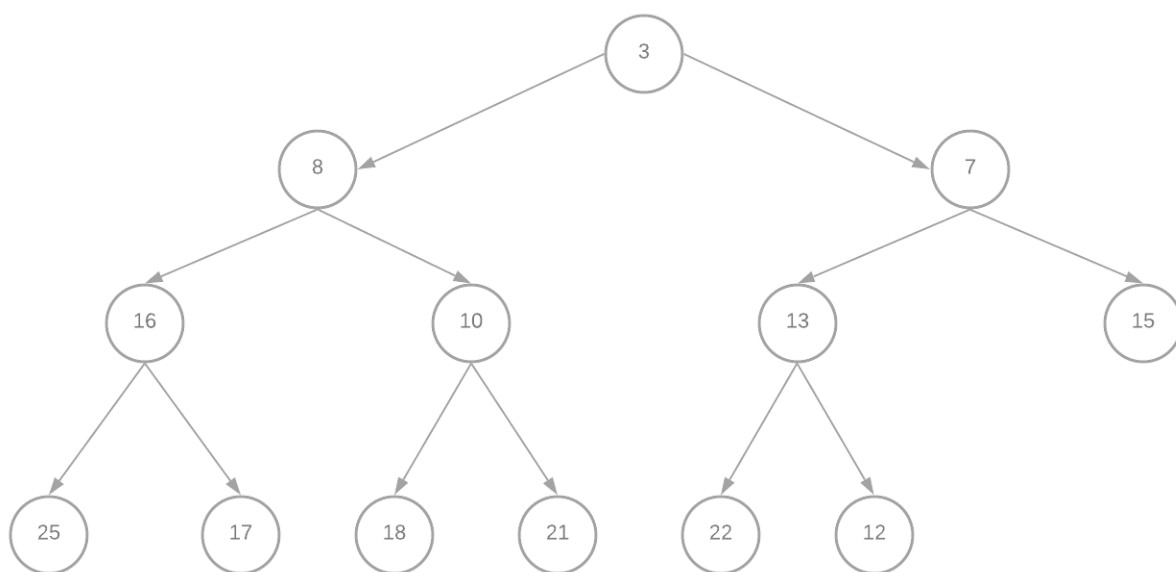


سپس مادامی که هر گره از هر دو فرزندش کوچکتر نشده است عمل bubble down را انجام میدهم. پایان پاک

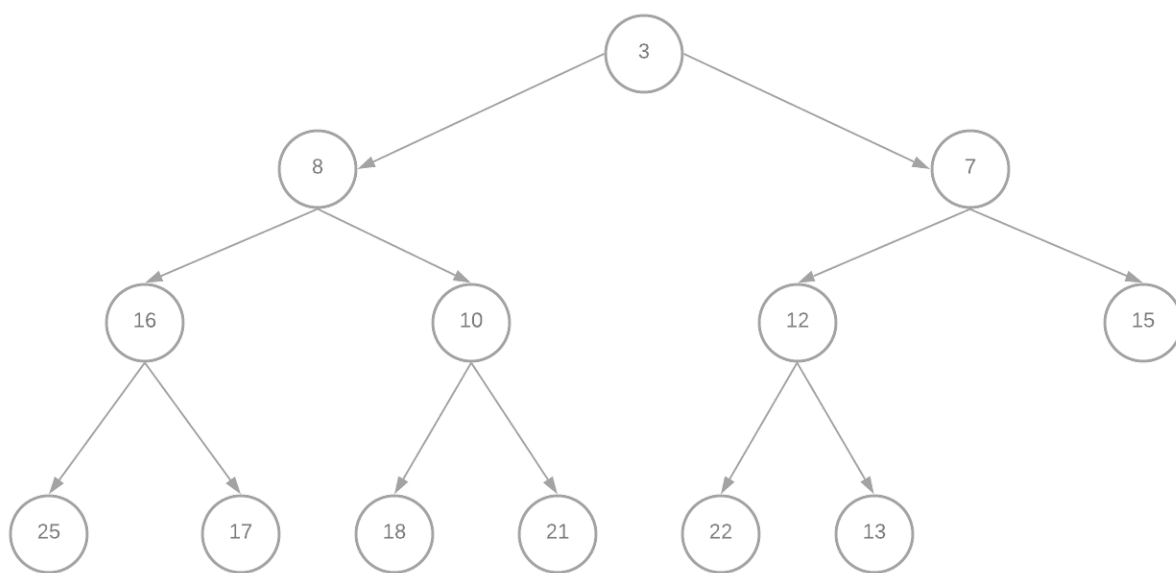
کردن 1.



ورود 12: ابتدا 12 را در آخرین جایگاه برگ های درخت قرار میدهیم.

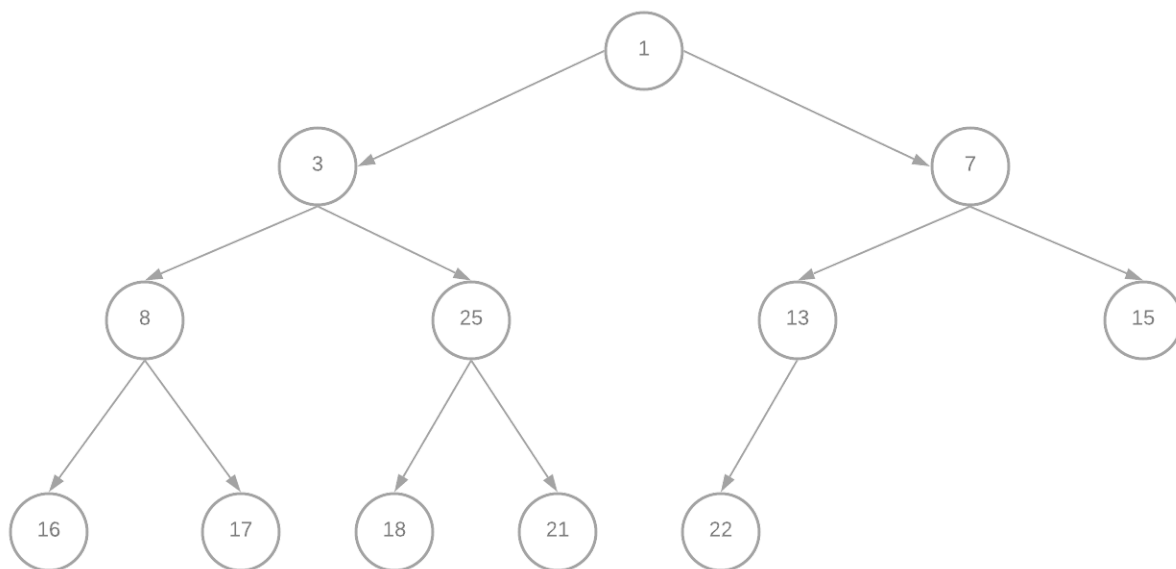


سپس مادامی که 20 از گره پدر خود کوچکتر است آن را bubble up میکنیم.

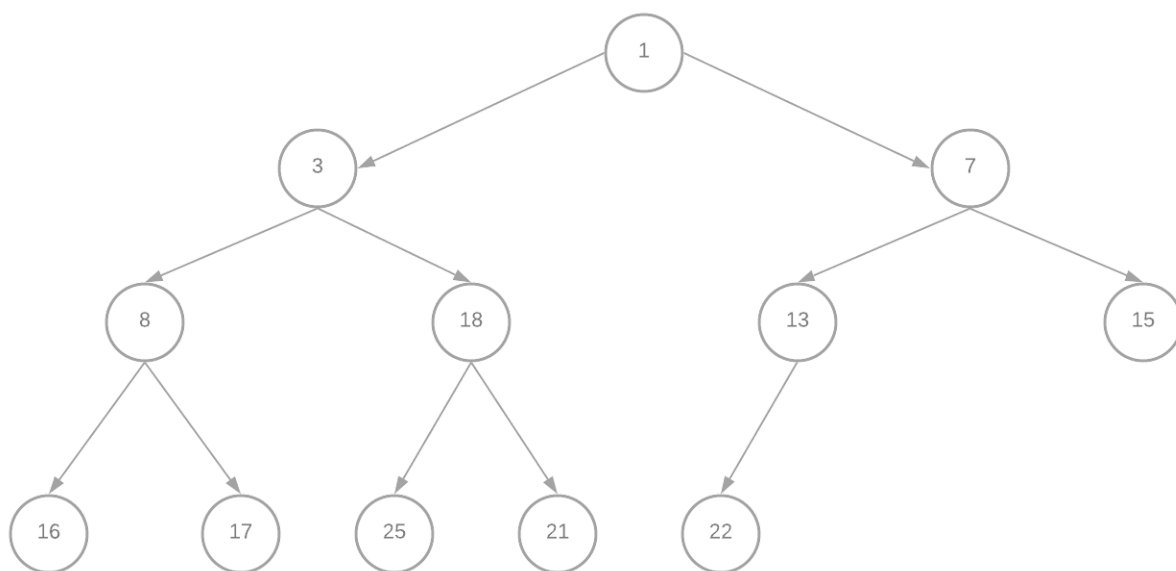


-3

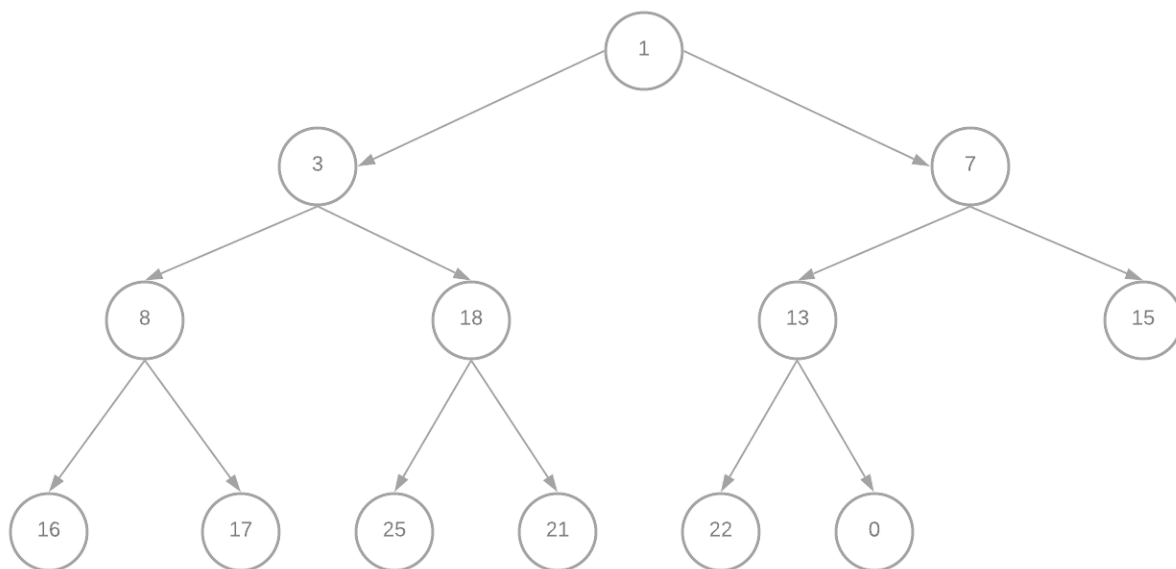
پاک کردن 10: ابتدا گره 10 را پاک میکنیم و آخرین گره درخت را جایگزین آن میکنیم.



حال مادامی که هر گره از درخت از پدر خود بزرگتر نیست، با توجه به وضعیت اعمال bubble و bubble up  
down را انجام میدهیم.



ورود 0: ابتدا 0 را در آخرین جایگاه خالی برگان درخت قرار میدهیم.



سپس مادامی که از گره پدر خود کوچکتر است، جای آن را با پدر خود عوض میکنیم.

