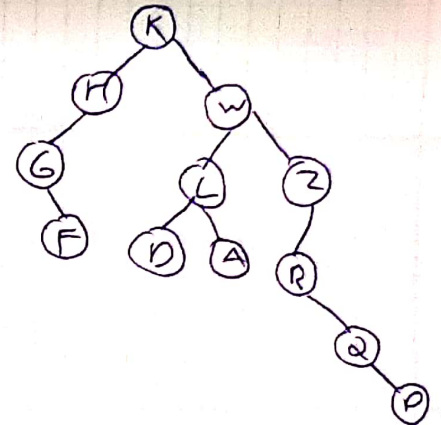


۹۸۷۶۵۴۳۲۱ ، مرتب شده است

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

L R V O F G H I J A L P Q R Z W K

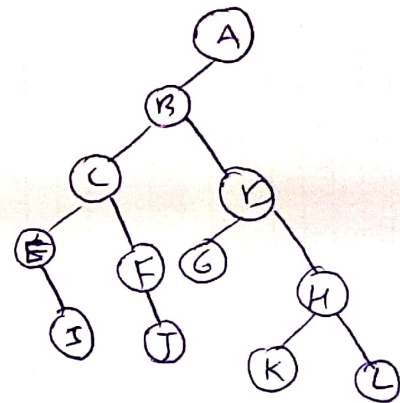
دسته



B C D E F G H I J K L M N O P Q R S T U V W X Y Z

L R V O F G H I J A L P Q R Z W K

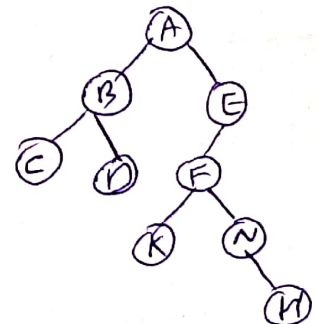
دسته



C D E F G H I J K L M N O P Q R S T U V W X Y Z

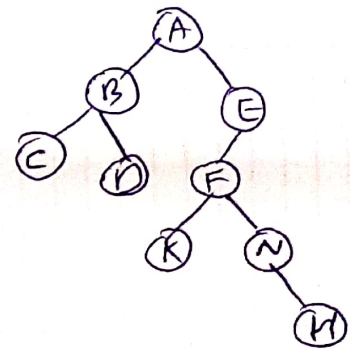
L R V O F G H I J A L P Q R Z W K

دسته



C B L R B C B D A K F N H E
 J L R B A B C D E F K N H

درخت



pre B A B C G H D E F K M I J L T
 post H G C E K F D B L J I T M A

سوال

الف) چه تعداد ریشه می توان کشید ؟

ب) چه تعداد برگ و گره های داخلی وجود دارد ؟

ج) صفی کشید

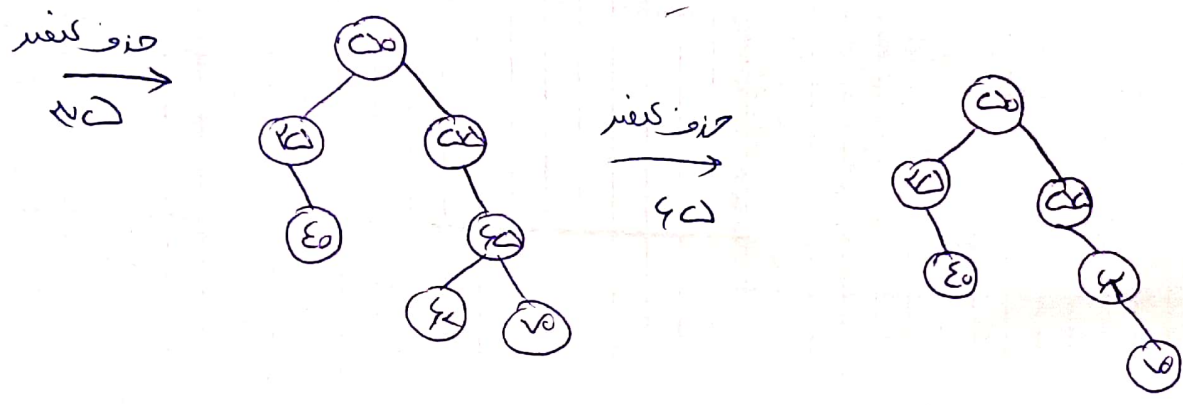
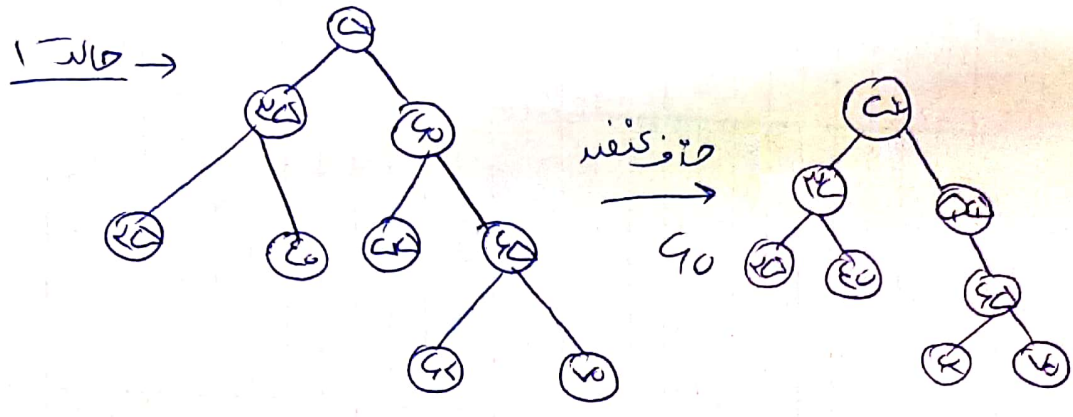
H و L و K و 6 ← به کدام حالت دارند

↓

$$2 \times 2 \times 2 = 8$$

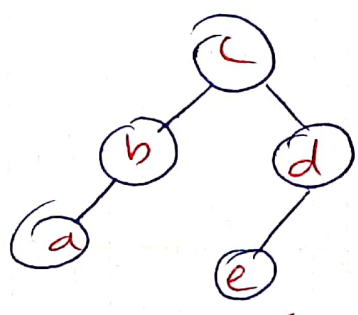
حالت

سوال ۴



سوال ۵

در دهایی e و d و c و b و a به این حالت می‌تواند وارد درخت T شود تا ریشه تا ریشه به ریشه به ریشه



فقط به حالت این می‌تواند
یعنی e و d از b بزرگتر هستند و
b و c و a از c کوچکترند

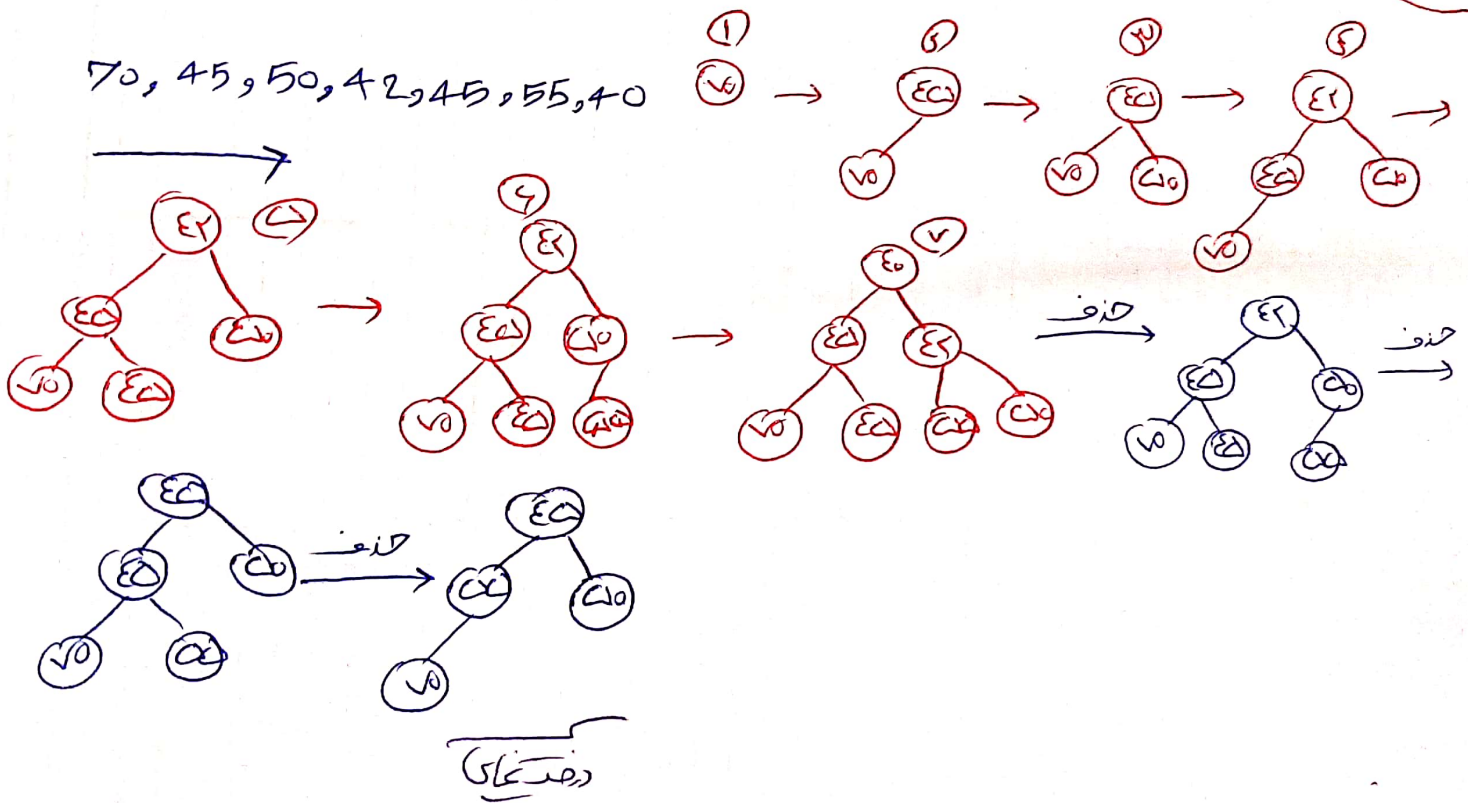
$e > d > c > b > a$
بزرگتر ←

binary search + BST Tree

در هر جستجو (دوروی) ← به هر راند باید بزرگتر از ریشه و به هر راند کوچکتر باید باشد ←
فقط به حالت این می‌تواند

سوال ۲
 Min heap خالی به ترتیب گره های با کدهای زیر اضافه کردیم. پس از ۳ عمل حذف، درخت؟

70, 45, 50, 42, 45, 55, 40



سوال ۶

آرایه زیر یک Max Heap است. برای راجع عدد 40 به گرهی که Heap باقی بماند و حین عمل exchange لازم است.

۷۵	۶۸	۷۴	۷۵	۷۷	۸۷	۸۷	۹۰	۱۰۰
----	----	----	----	----	----	----	----	-----

MAX heap است ← عدد اول ۷۵ در آرایه قرار میگیرد (با شروع از خانه ۱ ← در خانه ۱۰ قرار میگیرد)
 طبق رابطه $parent = \lfloor \frac{i}{2} \rfloor$ ← پس برای پیدا کردن اندیس parent
 باید حاصلش را یکی صغیر از ریشه کوچکتر به نسبت در همان خانه باقی میماند
 سپس با $x = \lfloor \frac{75}{2} \rfloor = 37$

۲ جایابی انجام دادیم

سؤال ۷ آرایه‌ای از n عدد از دسترس در اختیار داریم و قصد داریم از روی آن یک max_heap بسازیم

الف) روشی حفر را به صورت تکراره که توضیح دهید.

ب) آیا مرتب است؟ $O(n)$ انجام داری

for ($i = (\frac{n}{2}) - 1$; $i \geq 0$; $i--$)

$\text{maxheapify}(arr, n, i)$ و $\text{نوع با آخرین گره ای که بزرگ نیست}$

$\text{maxheapify}(\text{int } arr[], \text{int } n, \text{int } i)$ بزرگ‌ترین گره heap هستند $\frac{n}{2}$ نو در دسترس
 $\text{largest} = i$ ؛
 $\text{left} = 2 * i + 1$ ؛ یچین
 $\text{right} = 2 * i + 2$ ؛ یچین والد
 $\text{if} (L < n \ \&\& \ arr[L] > arr[\text{largest}])$
 $\text{largest} = L$ ؛ د فرزند دارند

$\text{if} (R < n \ \&\& \ arr[R] > arr[\text{largest}])$
 $\text{largest} = R$ ؛ در ارتفاع ها

$\text{if} (\text{largest} \neq i)$
 $\text{swap}(arr[i], arr[\text{largest}])$ ؛
 $\text{maxheapify}(arr, n, \text{largest})$ ؛
 $\}$

$$n \times O\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots\right) < 1$$

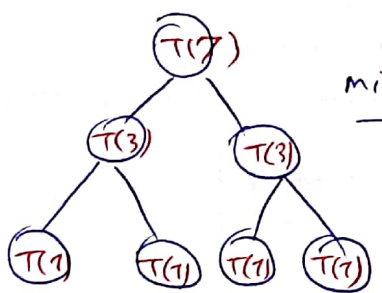
$$= O(n)$$

خوب سؤال ب

۷

سؤال ۸

با یکدیگر میزنند و چند نفر میزنند و باقی ماند



min-heap
یکی از گره ها که به عنوان root باید باشد
↓
و با یکدیگر میزنند

حالا همه عناصر از ریشه گرفته میزنند + مرتبه ای که میزنند
حالا از عناصر باقی مانده اینها را میزنند
 $(\frac{n}{2}) = 40$

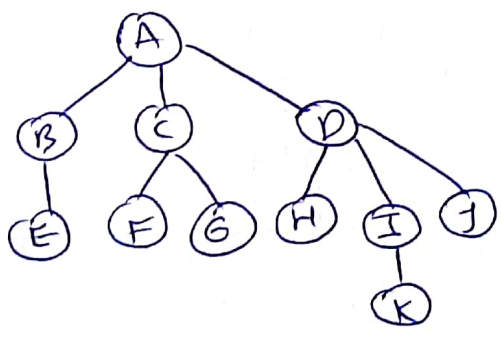
$T(n) = \binom{n-1}{1} * T(L) * T(R) \rightarrow$ به صورت بازگشتی دوباره
برای این ۳ عنصر یکی ریشه و دو تا بقیه

زیر درخت $\leftarrow 2 = (2)$ و بعد از این هم (2) مرتبه + در کل

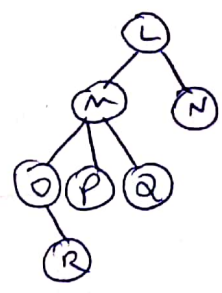
$40 < x < 80 =$ ۸۰
هفت جواب

سؤال ۹

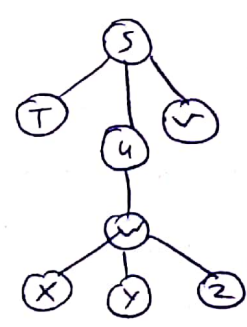
چند زیر درخت بزرگ



درخت اول

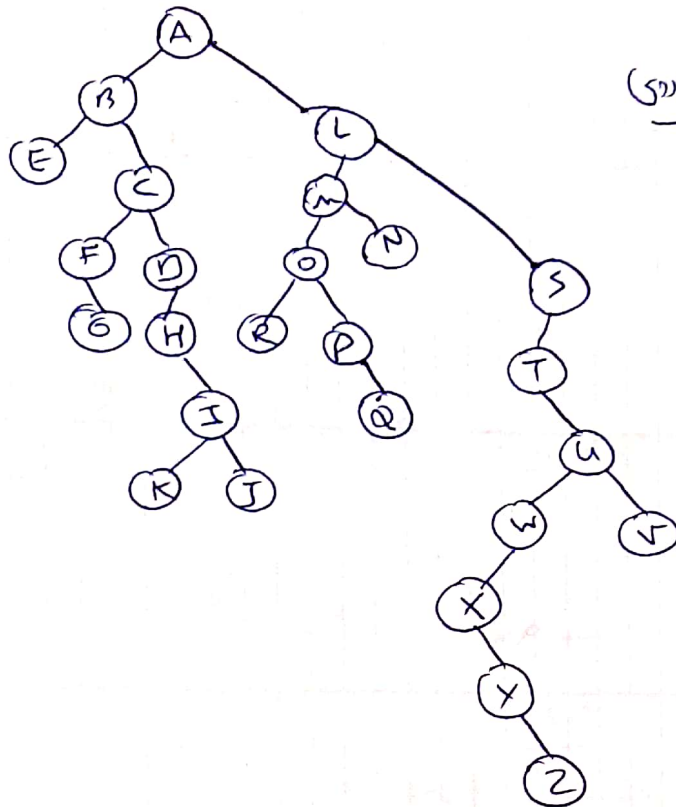


درخت دوم



درخت سوم

ب. با سه ادبی ۶ درخت را میزنند



تبدیل جنس به درخت دوری

اول ریشه درخت اول بعد از ریشه درخت اول است pre ①
درخت نهای ریشه

ب. (ب) صحیح ہے

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ✓

-pre \rightarrow اول نشانی بعد از پرانتز \rightarrow ABCDEFGHIKJLMORPQNST
 در عبارت را در $uvwxyz$
 \rightarrow درخت = pre = درخت = pre
 اینی نشانی

درفد دواوی
حنبل

→ پیش = pre = دفعہ = pre
پیش = پیش = پیش

سلسلہ

✓
ب۔ 48 (ج) حل

$-pre \rightarrow$ اءل نسا لعل زهر در دست كز
درفت ادوسى
عنى

$\rightarrow A B C F G D H I K J L M O R P Q N S T$
 $U W X Y Z V$

$\rightarrow -pre = درفت = pre$
بانبرى تكسى نه

ا

E B F G H K I J N A R O P Q M N L T u y z w u v s

EBFGCHKIJDAROPQMNLT → اول زید در رفت های حید → PRE = در رفت دور ی
 myz w u v s ← بعد از سه بعد زید در رفت های کد را زد ← باقیه را زد
 In = در رفت - In = در رفت - In
 In = در رفت - In

③ ^N post زیردفعه های اول بعد از دفعه های اول

سی / سی

E G F K J I H D C B R Q P O N M Z Y X W V U T S L A

اول دفعه می بعد از دفعه اول بعد از دفعه \rightarrow post - دفعه

E G F K J I H D C B R Q P O N M Z Y X W V U T S L A

پایه های pre و in و out می هستند اما post و order و اینها به این ترتیب