Data Structure

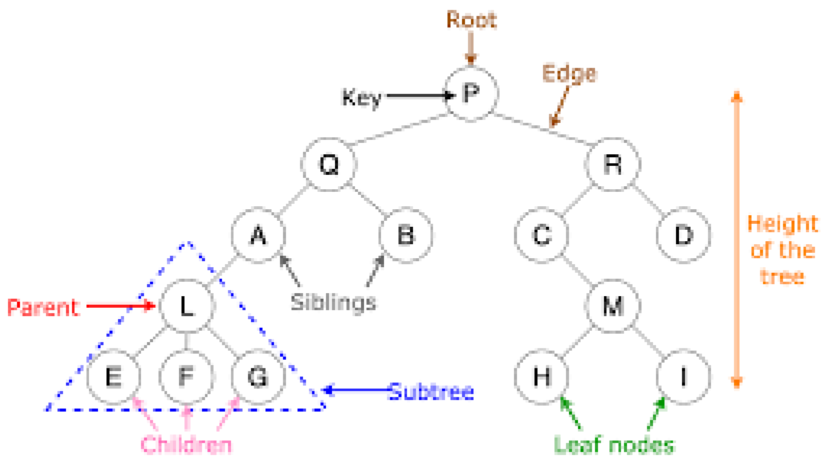
DR. RASTGOO

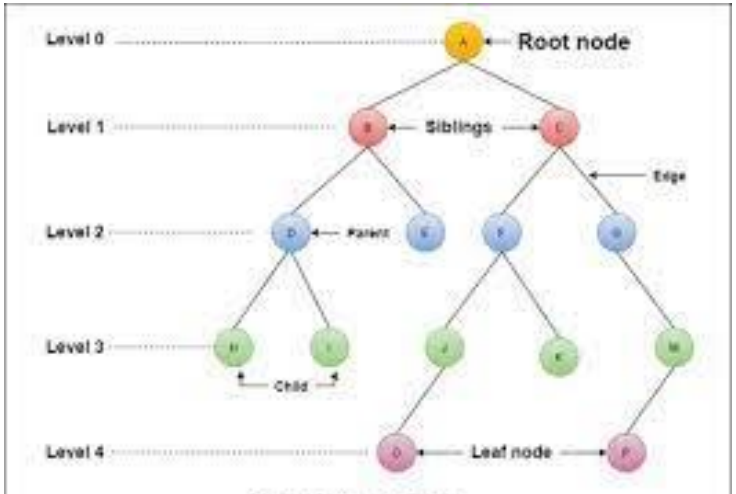
فصل ۶:

درخت

درخت

یک درخت مجموعه محدودی از یک یا چند گره می باشد که دارای گره خاصی به نام ریشه است و بقیه گره‌ها به n مجموعـه مجزا تقسیم می‌شوند که هر یک از مجموعه‌ها خود نیز یک درخت می باشند.



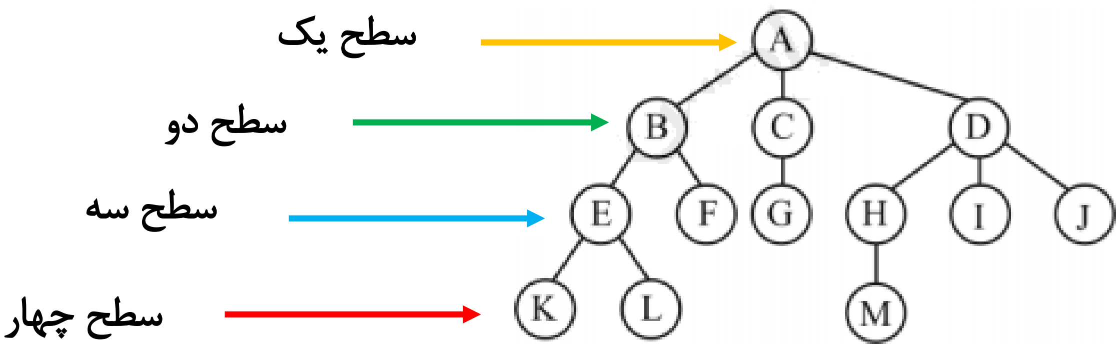


Tree Representation

درجه گره	تعداد زیردرخت های یک گره. (تعداد یالهایی که در گره برخورد دارند)
برگ	گره بدون فرزند (گره با درجه صفر)
ارتفاع گره V	طول بزرگترین مسیر از V به برگ W به طوری که W گره ای از زیر درختی بـه ریشـه V باشد.
ارتفاع درخت	ارتفاع ریشه
سطح(یا عمق) یک گره	طول مسیر از آن گره تا ریشه
درخت متوزان	درختی که سطح برگهای آن حداکثر یک واحد اختلاف داشته باشد.
درخت کاملا متوزان	درختی که سطح برگهای آن برابر است.

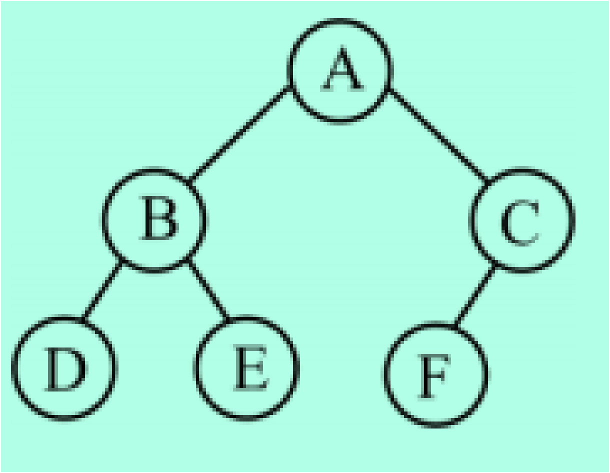
مثال

ارتفاع درخت زیر ۴ است: (ریشه را در سطح یک فرض کرده‌ایم)



درخت دودویی

درختی که تعداد فرزندان هر گره در آن حداکثر برابر دو باشد را درخت دودویی می نامند. به عبارتی یک درخـت دودویـی یـا تهی است و یا حاوی مجموعه‌ای محدود از گره‌ها، که هر گره حداکثر دو فرزند دارد. مانند درخت زیر:



بیشترین تعداد گره‌ها روی سطح i ام یک درخت دودویی برابر $2^{(i-1)}$ می باشد.

بیشترین تعداد گره‌ها در یک درخت دودویی به ارتفاع h، برابر $2^h - 1$ می باشد.

مثال

بیشترین تعداد گره‌ها روی سطح پنجم یک درخت دودویی چند است؟ (ریشه در سطح یک می باشد.)

پاسخ: بیشترین تعداد گره‌ها روی سطح پنجم یک درخت دودویی، طبق رابطه $2^{(i-1)}$ برابر 16 است.

مثال

بیشترین تعداد گره‌ها در یک درخت دودویی به ارتفاع 4 چند است؟

پاسخ: بیشترین تعداد گره‌ها در یک درخت دودویی به ارتفاع 4 ، طبق رابطه $2^h - 1$ برابر 15 است.

در هر درخت دودویی رابطه $n_0 = n_2 + 1$ برقرار است. (n_0 تعداد برگها و n_2 تعداد گره‌های دو فرزندی)

مثال

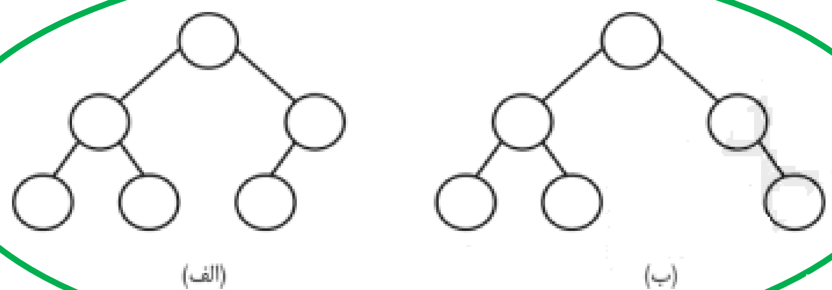
تعداد برگها در یک درخت دودویی شامل 10 گره با درجه دو را بدست آورید؟

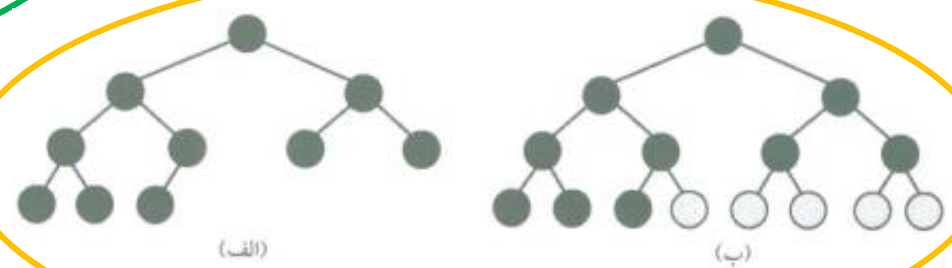
پاسخ: با توجه به رابطه $n_0 = n_2 + 1$ ، تعداد برگها برابر 11 می باشد.

درخت کامل

درختی که تمام سطح‌های آن بجز احتمالاً آخرین سطح، حداکثر تعداد گره‌های ممکن را داشته باشد و گره‌های سطح آخر تا حد امکان در سمت چپ باشند را کامل می نامند. به طور نمونه درخت شکل الف کامل می باشد ولی درخت شکل ب کامل نمی‌باشد:

نمونه ای از درخت دودویی کامل و تبدیل آن به درخت پر:





ارتفاع درخت دودویی کامل برابر $\lfloor \log_2^n \rfloor + 1$ می باشد.($\lfloor \log_2^n \rfloor + 1$ همان حد بالای $\log_2^{(n+1)}$ می باشد.)

تعداد گره های سطح آخر یک درخت دودویی کامل با n گره برابر است با :

$n-(2^{(h-1)}-1)$

درخت پر

درختی که هم کامل و هم کاملاً متوازن باشد را درخت پر می نامند. در این درخت:

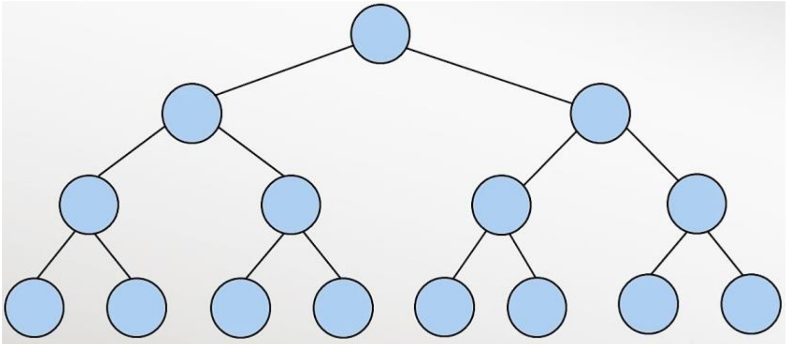
۱- ارتفاع درخت برابر $\log_2^{(n+1)}$ می باشد.

۲- تعداد گرهها برابر $2^h - 1$ می باشد.

۳- تعداد برگها، برابر $\dfrac{n+1}{2}$ و یا برابر $2^{(h-1)}$ می باشد.

۴- تعداد گره ها در سطح i ، برابر $2^{(i-1)}$ می باشد.

۵- تعداد لینک ها یکی کمتر از تعداد گره ها می باشد.



کران پایین و کران بالای تعداد گره ها

در یک درخت دودویی، کران بالای n برابر 2^h-1 (درخت پر) و کران پایین n برابر h می باشد. (درخت اریـب).

درخت دودویی	$h \le n \le 2^h - 1$
درخت دودویی کامل	$2^{(h-1)} \le n \le 2^h - 1$
درخت دودویی پر	$2^h - 1 \le n \le 2^h - 1$

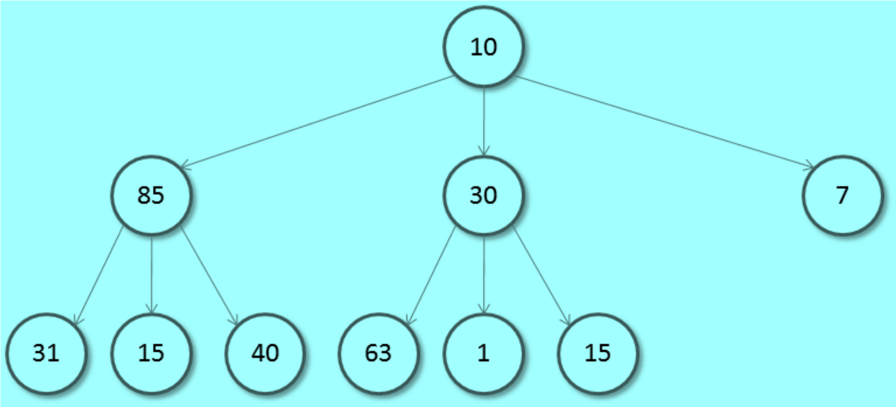
کران بالای ارتفاع درخت برابر n و کران پایین آن برابر $\lfloor \log n \rfloor + 1$ می باشد. بنابراین می توان نوشت:

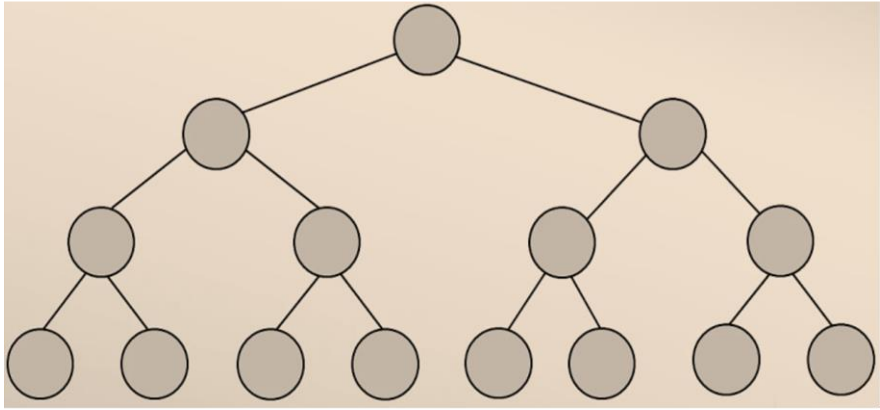
$$\lfloor \log n \rfloor + 1 \le h \le n$$

با n گره می توان 2^{n-1} درخت دودویی متمایز(از لحاظ توپولوژی) به ارتفاع n ساخت.

درخت k تایی

درختی که حداکثر تعداد فرزندان هر گره آن برابر k باشد را درخت k تایی می نامند. معروفترین درخت k تایی، درخت دوتایی (دودویی) است.





در درخت k تایی با n گره تعداد nk اتصال وجود دارد که n-1 اتصال استفاده شده و nk-(n-1) اتصال تهی می باشد.

تعداد برگ ها در یک درخت k تایی :

$$n_0 = (k-1)n_k + (k-2)n_{k-1} + \ldots + n_2 + 1$$

(n_0 : تعداد برگها) (n_k : تعداد گره های k فرزندی)

تعداد برگ ها مستقل از تعداد گره های تک فرزندی است.

مثال

در یک درخت سه تایی، اگر 5 گره دو فرزندی و 10 برگ داشته باشیم، آنگاه چند گره 3 فرزندی خواهیم داشت؟

پاسخ: در یک درخت سه تایی، رابطه زیر برقرار است:

$$\boxed{n_0 = 2n_3 + n_2 + 1 \Rightarrow 10 = 2n_3 + 5 + 1 \Rightarrow n_3 = 2}$$

تعداد برگ ها در یک درخت **k تایی کامل** با n گره، برابر است با: $\left\lfloor \frac{(k-1)n+1}{k} \right\rfloor$

(یا $n-\left\lceil \frac{n-1}{k} \right\rceil$)

تعداد گره های غیر برگ در یک درخت **k تایی کامل** با n گره، برابر است با : $\left\lceil \frac{n-1}{k} \right\rceil$

با دادن مقادیر ۲و۳و۴ به پارامتر k ، جدول زیر به دست می آید:

تعداد برگها در درخت دوتایی کامل	$n_0 = \left\lfloor \frac{n+1}{2} \right\rfloor$
تعداد برگها در درخت سه تایی کامل	$n_0 = \left\lfloor \frac{2n+1}{3} \right\rfloor$
تعداد برگها در درخت چهارتایی کامل	$n_0 = \left\lfloor \frac{3n+1}{4} \right\rfloor$

مثال

یک درخت چهار تایی کامل با 27 گره، دارای چند گره برگ می باشد؟

$$n_0 = \left\lfloor \frac{3n+1}{4} \right\rfloor = \left\lfloor \frac{3\times 27+1}{4} \right\rfloor = \lfloor 20.5 \rfloor = 20$$

مثال

حداکثر تعداد گره ها در یک درخت دودویی با **b** برگ چقدر است؟

پاسخ: کران بالا ندارد. چون یک درخت دودویی می تواند یک مسیر باشد و بی نهایت گره و تنها یک برگ داشته باشد.

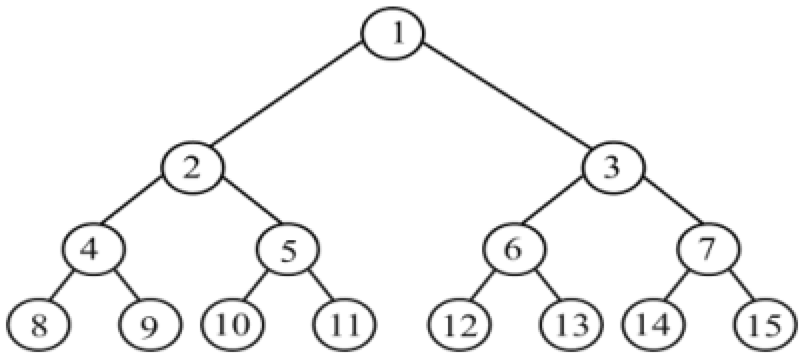
بیشترین تعداد گره‌ها روی سطح i ام یک درخت k تایی برابر است با : $k^{(i-1)}$

بیشترین تعداد گره‌ها در یک درخت k تایی به ارتفاع h، برابر است با : $\frac{k^{h}-1}{k-1}$

تعداد برگها در درخت k تایی پر به ارتفاع h برابر است با : $k^{(h-1)}$

درخت دودویی کامل شماره گذاری شده

به هر یک از گره های یک درخت دودویی کامل می توان یک شماره نسبت داد.



یک درخت دودویی کامل شماره گذاری شده با ارتفاع ۴

در هر درخت دودویی کامل شماره گذاری شده، برای هر گره با اندیس i قواعد زیر برقرار است:

۱) اگر $i=1$ باشد، آنگاه ریشه است.

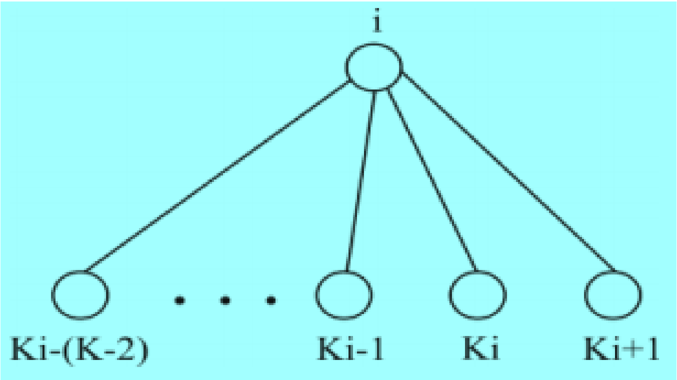
۲) اگر $i<>1$ ، آنگاه والد آن یعنی parent(i) ، در $\left\lfloor \frac{i}{2} \right\rfloor$ است.

۳) اگر $2i<=n$، آنگاه فرزند چپ آن یعنی Leftchild(i) ، در $2i$ است.

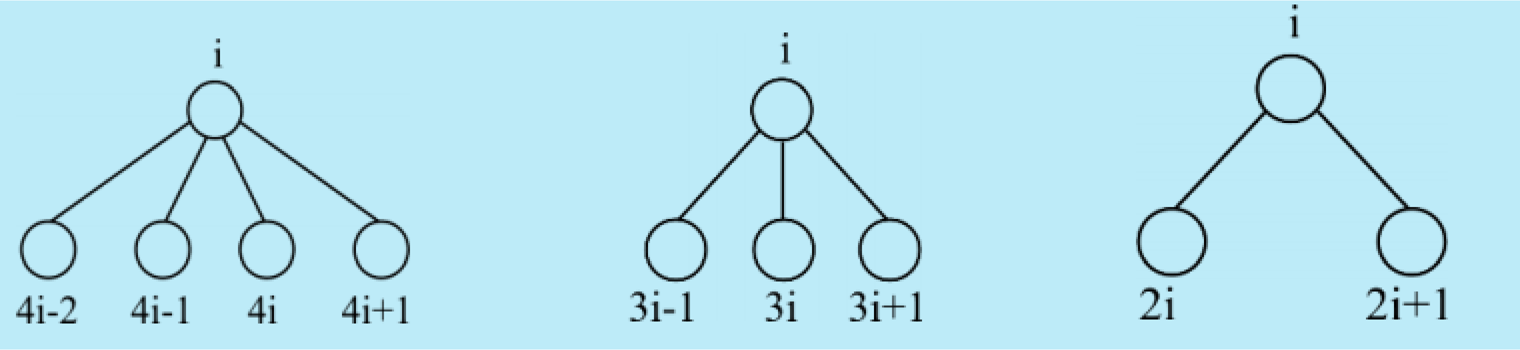
۴) اگر $2i+1<=n$ ، آنگاه فرزند راست آن یعنی Rightchild(i) ، در $2i+1$ است.

تذکر: اگر 2i>n ، آنگاه i فرزند چپ ندارد و اگر 2i+1>n باشد، آنگاه i فرزند راست ندارد.

یک گره با شماره i در یک درخت k تایی دارای فرزندان با شماره های زیر می باشد:



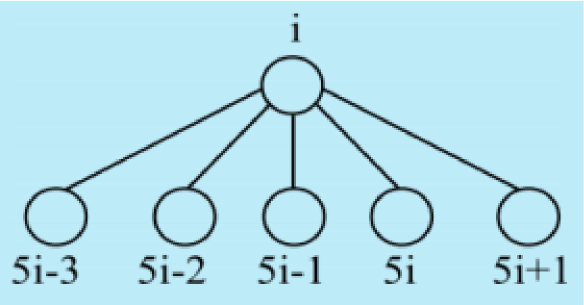
به طور نمونه در درخت های ۲ تایی، ۳ تایی و ۴ تایی داریم:



مثال

گره با شماره 15، در یک درخت 5 تایی کامل، پدر کدام گره ها می باشد؟(n=95)

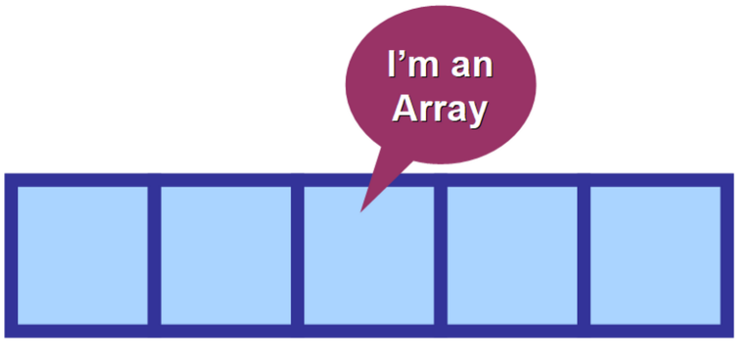
پاسخ: گره i ام پدر گره های 5i+1 , 5i , 5i-1 , 5i-2 , 5i-3 می باشد:

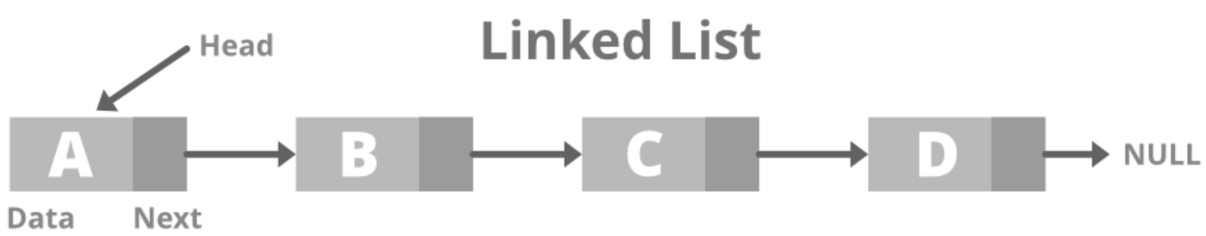


بنابراین گره پانزدهم پدر گره های 76 , 75 , 74 , 73 , 72 می باشد.

روشهای ذخیره درخت دودویی

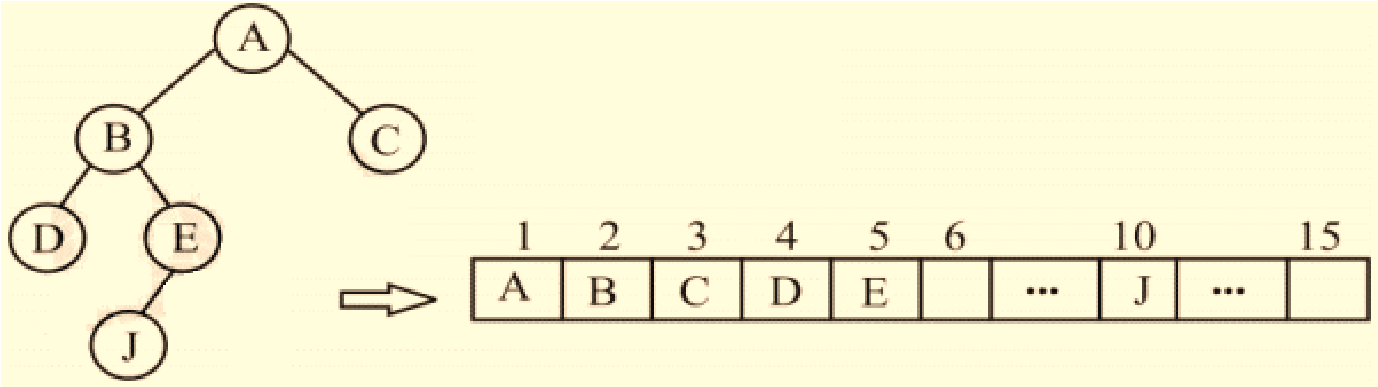
یک درخت دودویی را می توان به کمک آرایه و یا لیست پیوندی در حافظه ذخیره کرد.





ذخیره درخت دودویی با استفاده از آرایه

برای ذخیره یک درخت دودویی به کمک آرایه ، گره i ام با توجه به درخت دودویی کامل شماره گذاری شـده ، در خانـه i ام در آرایه ذخیره می شود. در شکل زیر ذخیره درخت داده شده در آرایه نمایش داده شده است:



گره با شماره 1 (با مقدار A) در خانه اول و گره با شماره 5 (با مقدار E) در خانه پنجم آرایه ذخیره می شود

ذخیره درخت دودویی با استفاده از لیست پیوندی

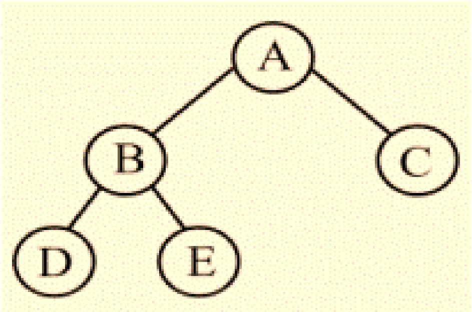
هر گره در درخت دودویی شامل سه قسمت "داده، اشاره گر به فرزند چپ و اشاره گر به فرزند راست " می باشد. تعریف یک گره درخت دودویی در زبان c به صورت زیر است:

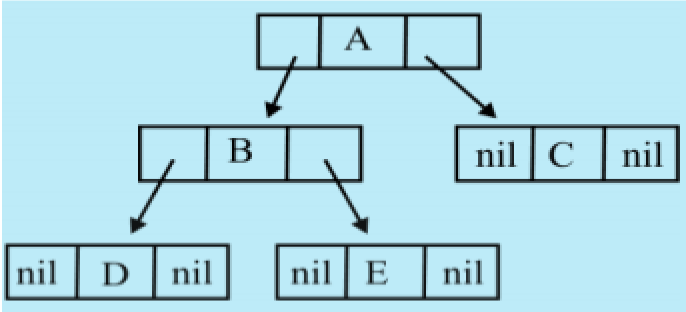
```
typedef struct node *tree-pointer;
typedef struct node{
    int            data;
    tree-pointer   left;
    tree-pointer   right;
};
```

درخت دودویی با n گره دارای 2n اشاره گر است که تعداد n-1 اشاره گر استفاده شده و n+1 اشاره گر استفاده نشده (تهی) می باشد.

مثال

نمایش درخت دودویی با استفاده از لیست پیوندی:





مشاهده می شود که درخت دودویی بالا با 5 گره، شامل 10 اشاره گر است که 4 اشـاره گـر آن اسـتفاده شـده و 6 اشـاره گـر استفاده نشده (nil) می باشد.

تعداد درخت های دودویی

تعداد زیر درخت های چپ ریشه

تعداد زیر درخت های چپ ریشه

$$T(n) = \sum_{i=1}^{n} T(i-1)T(n-i)$$

$$T(0) = T(1) = 1$$

تذکر: این رابطه را می توان به صورت $b_n = \sum_{k=1}^{n} b_{k-1} b_{n-k}$ نیز نشان داد. مقدار b_n برابر $\frac{1}{n+1}\binom{2n}{n}$ می باشد که همان n امین عدد کاتالان است. می توان نشان داد که :

$$b_n = \frac{4^n}{\sqrt{\pi}\ \ n^{3/2}}(1 + O(1/n)) \in \Omega(2^n)$$

الگوریتم های کار بر روی درخت دودویی

کپی گرفتن از درخت دودویی

```
tree-pointer  copy (treepointer  *p){
    tree-pointer  temp;
    if (p){
            t = malloc(sizeof(node) );
            t -> data   = p -> data;
            t -> lchild = copy (p  -> lchild );
            t -> rchild = copy (p  -> rchild);
           return  t ;
      }
     return (NULL);
}
```

الگوریتم های کار بر روی درخت دودویی

محاسبه ارتفاع درخت دودویی

```
int  depth (tree-pointer   root ) {
   int  d;
   if ( root == NULL)  return  0;
   x = depth( root  − > left  );
   y = depth( root  − > right );
   if  (x > y)
         d=x+1;
   else
        d=y+1;
   return  d;
 }
```

الگوریتم های کار بر روی درخت دودویی

شمارش تعداد نودهای درخت دودویی

```
int  count (tree-pointer  root ){
   int  c;
   if ( root== NULL)  return  0;
   a =  count ( root  – > left ) ;
   b = count ( root  – > right );
   c = a+b+1;
   return c;
}
```

الگوریتم های کار بر روی درخت دودویی

شمارش تعداد برگهای درخت دودویی

```
int  f (treepointer  *t){
     if  (t==NULL)  return  0;
     else  if (t−>lchild==NULL)  && ( t−>rchild==NULL)
              return 1;
          else
            return ( f (t−>lchild) + f (t−>rchild) );
}
```

می توان این تابع را به صورت زیر نشان داد:

$$f(t) = f(left(t]) + f(right[t]) + IsLeaf(t)$$

که تابع IsLeaf(t) اگر t برگ باشد مقدار 1 و گرنه مقدار 0 را برمی گرداند.

پیمایش درخت دودویی

پیمایش درخت، یعنی حرکت روی یالهای درخت و ملاقات همه گره های آن دقیقا یکبار. اگـر انشـعاب بـه چـپ در هـر گـره درخـت دودویـی را بـا L ، انشـعاب بـه راسـت را بـا R و ملاقـات گـره را بـا V نمـایش دهـیم، امکـان تولیـد شـش ترکیـب RLV,RVL,VRL,LRV,LVR,VLR وجود دارد.

در سه پیمایش همواره انشعاب به چپ قبل از انشعاب به راست صورت گرفته که روشهای معمول می‌باشند و عبارتند از:

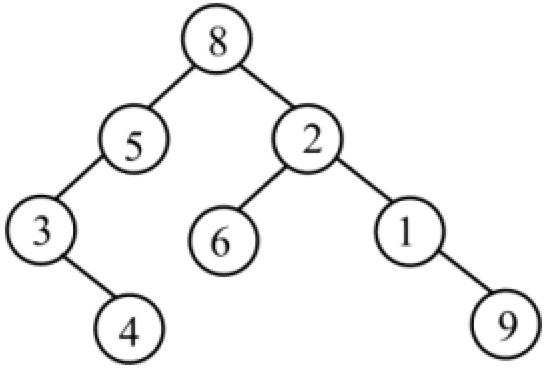
۱) پیشوندی (VLR) (ریشه ـ چپ ـ راست) (preorder)

۲) میانوندی (LVR) (چپ ـ ریشه ـ راست) (inorder)

۳) پسوندی (LRV) (چپ ـ راست ـ ریشه) (postorder)

مثال

حاصل پیمایش های معمول درخت زیر را بدست آورید.



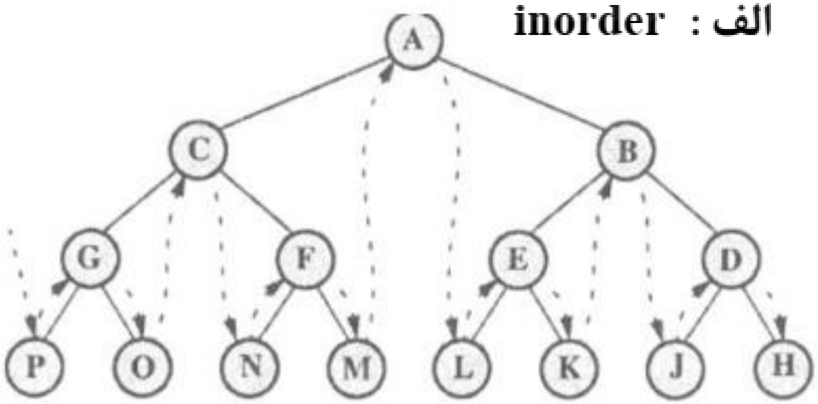
inorder : 34586219

preorder : 85342619

postorder : 43569128

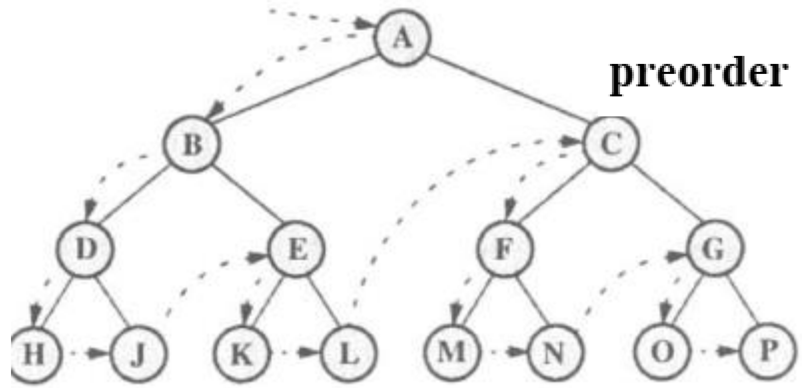
مثال

الف : inorder



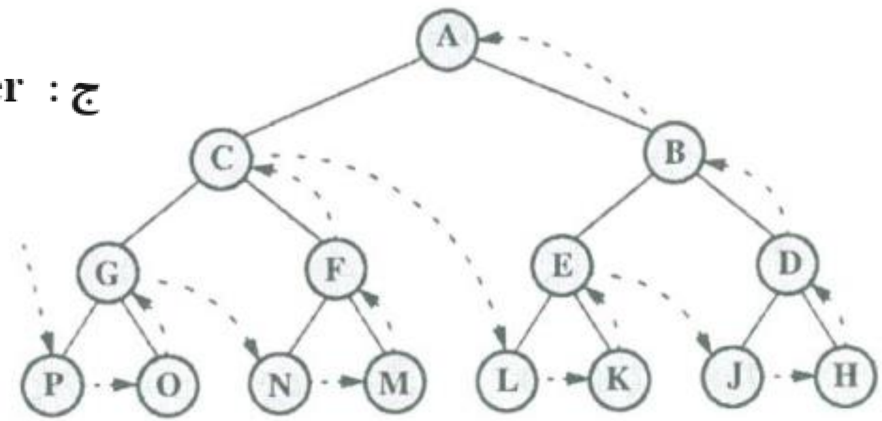
PGOCNFMALEKBJDH

ب : preorder



ABDHJEKLCFMNGOP

ج : postorder



POGNMFCLKEJHDBA

رسم درخت دودویی با داشتن دو پیمایش

می توان با داشتن دو پیمایش از یک درخت دودویی، درخت را ترسیم کرد، به شرط آنکه یکی از آنها میانوندی باشد.

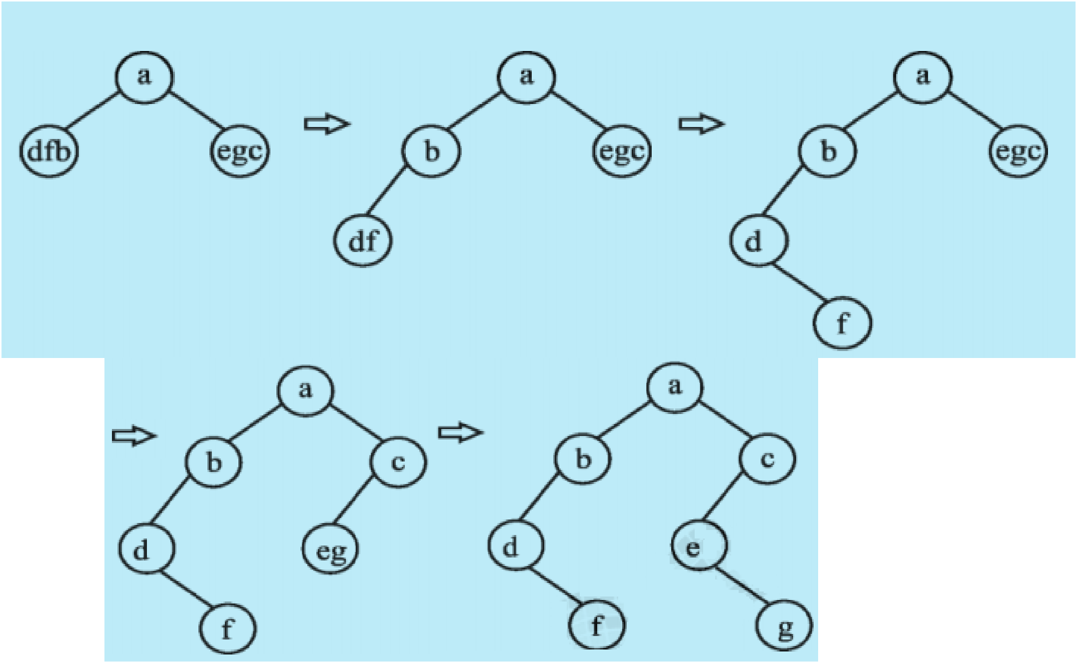
مثال

درخت دودویی که پیمایش پیشوندی آن abdfceg و پیمایش میانوندی آن dfbaegc باشد، را رسم نمایید.

گره a ریشه است، چون ریشه اولین گره از پیمایش پیشوندی است

گره dfb در سمت چپ ریشه

گره های egc در سمت راست ریشه



با داشتن پیمایش postorder یک درخت دودویی کامل با برچسب‌های متفاوت و مشخص بودن برچسبهای برگهای درخت، می‌توان درخت را ساخت و حاصل درختی واحد است.

اگر پیمایشهای پیشوندی و پسوندی یک درخت دودویی با n گره در دسترس باشند، ریشه و گره‌های تک فرزندی را می‌توان تعیین کرد، اما محل گره‌های تک فرزندی را نمی‌توان مشخص کرد. بنابراین در صورت وجود گره های تک فرزندی، چندین درخت می توان ایجاد کرد که پیمایش preorder و postorder آنها با هم برابر باشند. تعداد این درخت ها برابر است با 2^k که k تعداد گره های تک فرزندی است.

نکته

در درخت دودویی با داشتن هر سه شرط زیر می توان درخت را ساخت و حاصل نیز یکتا می باشد:

۱- همه گره ها برگ یا دو فرزندی باشند.

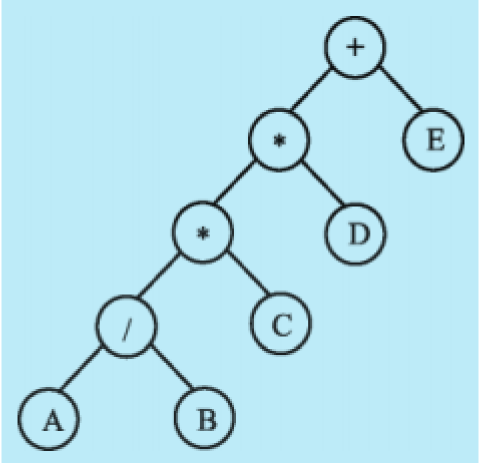
۲- برچسب برگ ها در پیمایش مشخص باشد.

۳- یکی از پیمایش های preorder یا postorder یا levelorder وجود داشته باشد.

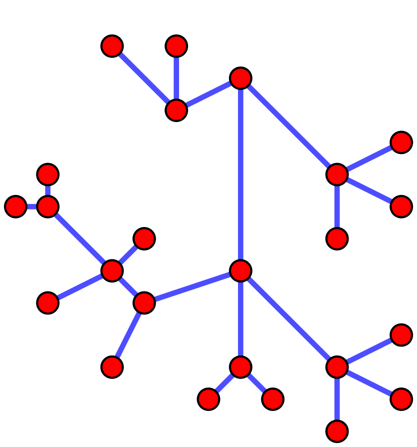
پیمایش درخت به ترتیب سطح

پیمایشهای میانوندی، پیشوندی و پسوندی به هر دو صورت بازگشتی یا تکرار مراحل، به پشته نیاز دارند. در پیمـایش درخـت به ترتیب سطح از صف استفاده می شود. در این پیمایش اول ریشه، بعد فرزند چپ ریشه و سپس فرزند راست ریشـه ملاقـات می شوند. این فرآیند تکرار می شود، بررسی گره ها در هر سطح جدید از سمت چپ به راست، صورت می گیرد.

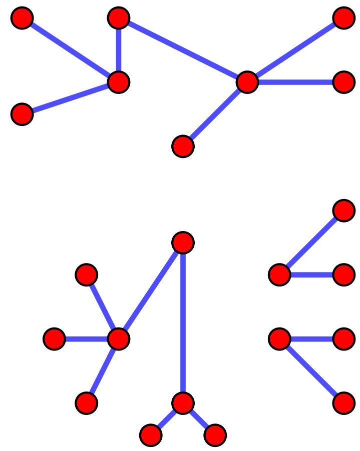
پیمایش به ترتیب سطح درخت زیر +*E*D/CAB می باشد:



جنگل



Tree



Forest

جنگل شامل n درخت مجزا است.

به عبارتی مجموعه‌ای مرتب از صفر یا چند درخت متمایز است.

بـا برداشتن ریشه یک درخت دودویی T، جنگل F به دست می‌آید.

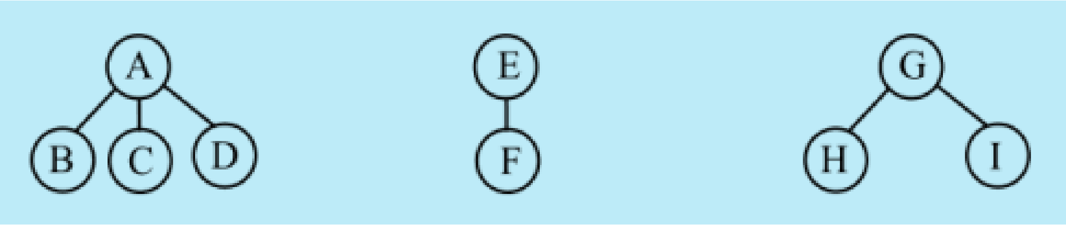
مراحل تبدیل جنگل به یک درخت دودویی عبارت است از:

۱- هر درخت جنگل را به یک درخت دودویی تبدیل می‌کنیم.

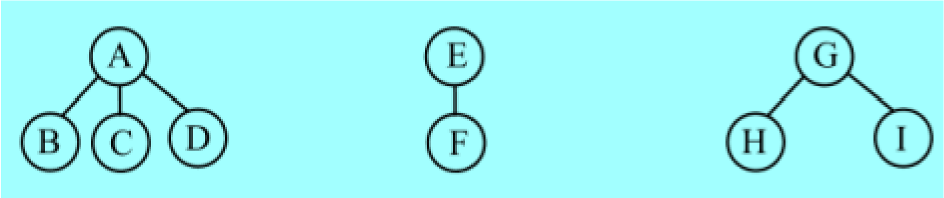
۲- درختهای دودویی را از طریق فرزند راست گره‌های ریشه به هم متصل می‌نماییم.

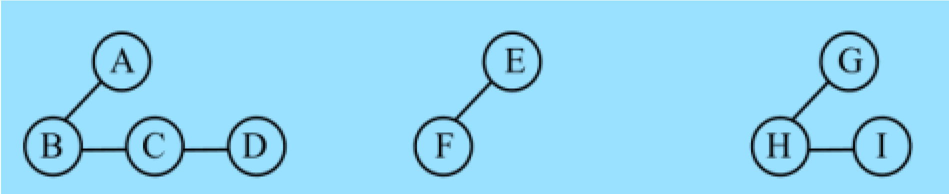
مثال

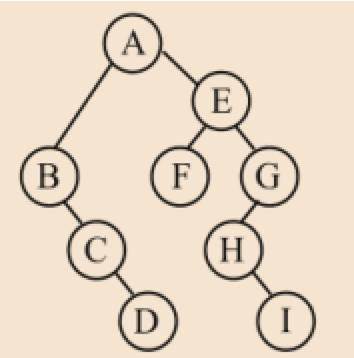
جنگل زیر که از ۳ درخت عمومی تشکیل شده را به یک درخت دودویی تبدیل کنید.



ابتدا هر یک از درختان عمومی را به دودویی تبدیل کرده و سپس درختهای دودویی را از طریق فرزند راست به یکدیگر متصل می نماییم.

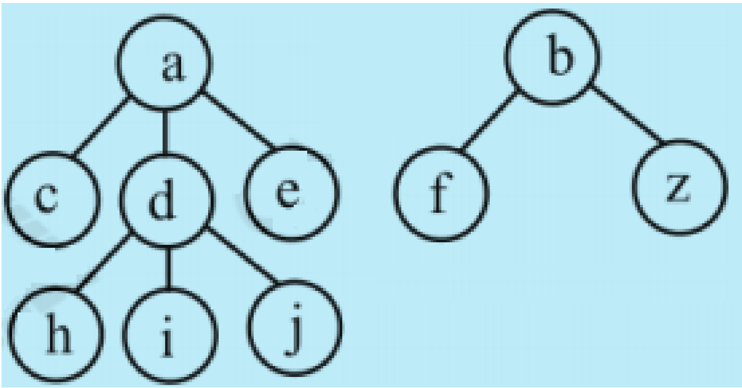

A
B
C
D
E
F
G
H
I


A
B
C
D
E
F
G
H
I

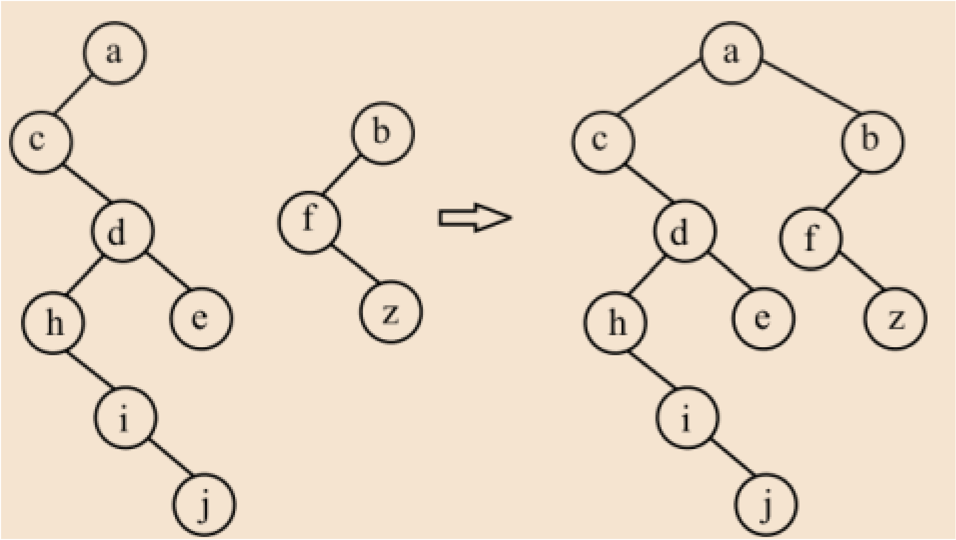

A
E
B
F
G
C
H
D
I

مثال

پیمایش inorder جنگل کدام است؟



ابتدا جنگل را به درخت دودویی تبدیل می کنیم :



پیمایش inorder درخت دودویی حاصل

chijdeafzb

پیمایش inorder درخت دودویی و جنگل نتایج یکسانی دارند.

پیمایش preorder درخت دودویی و جنگل نتایج یکسانی دارند.

پیمایش postorder درخت دودویی و جنگل همیشه نتایج یکسانی ندارند.