



داده ساختار و الگوریتم

Data Structures

میلاذ سلطانی

فهرست مطالب فصل پنجم

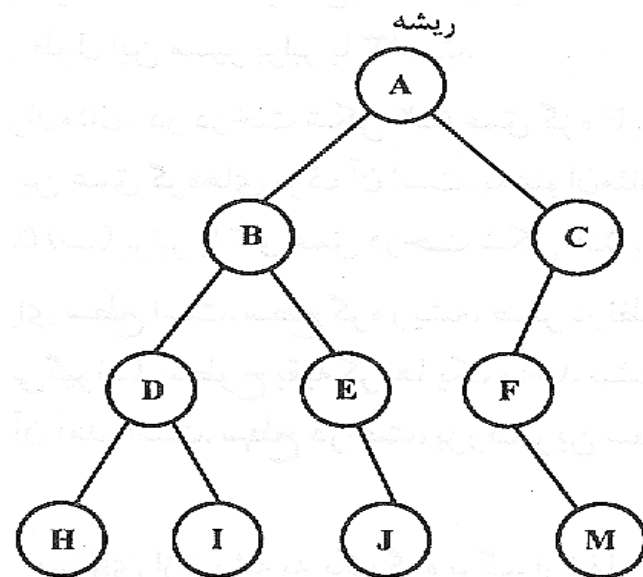
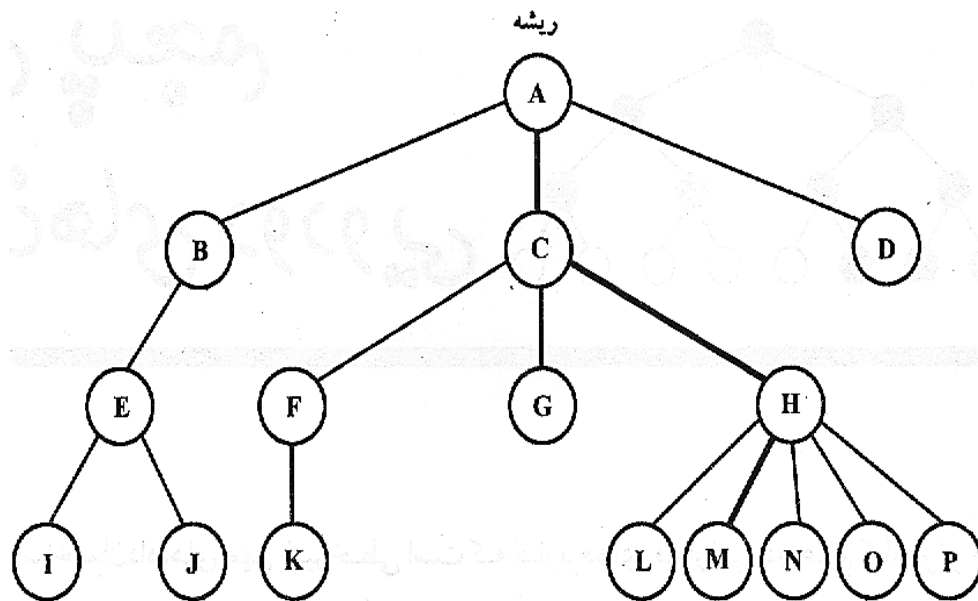
- مقدمه
- اصطلاحات درخت‌ها
- پیاده سازی درخت دودویی
- آرایه
- لیست پیوندی
- انواع پیمایش درخت دودویی
 - پیشوندی
 - میانوندی
 - پسوندی

درخت‌ها

Trees

مقدمه

- درخت یکی از ساختمان داده های مهم غیر خطی است که در علم کامپیوتر کاربردهای فراوانی دارد.
- بر خلاف درخت های طبیعی، در درخت های کامپیوتری، ریشه در بالا و برگ ها در پایین قرار دارند.



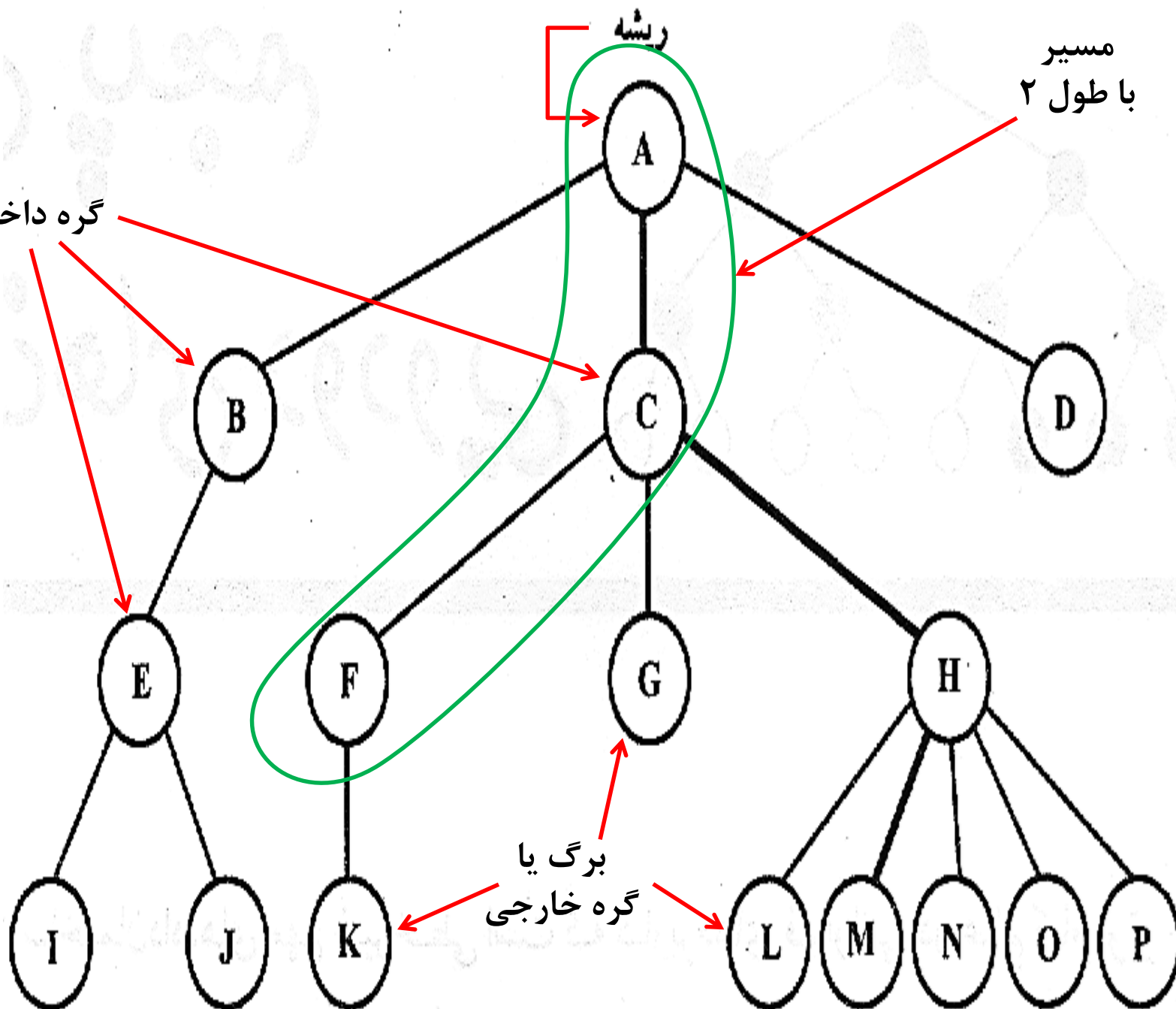
اصطلاحات مربوط به درخت ها

- **گره:** عناصر درخت را گره می گویند.
- **گره برگ:** گره هایی که هیچ فرزندی ندارند، برگ نامیده می شوند.
- **گره ریشه:** بالاترین گره درخت، ریشه نام دارد.
- **گره داخلی:** به تمام گره های غیر برگ، گره داخلی گفته می شود.
- **گره خارجی:** نام دیگر گره های برگ، گره خارجی است.
- **مسیر:** دنباله ای از گره های همجوار را مسیر می گویند.
- **طول مسیر:** تعداد اتصال های همجوار در یک مسیر یا یک واحد کمتر از تعداد گره های موجود در مسیر است.

گره داخلی

ریشه

مسیر
با طول ۲



اصطلاحات مربوط به درخت ها

- **عمق گره:** به طول مسیر یک گره تا ریشه گفته می شود.
- **عمق درخت:** به بزرگترین عمق بین گره های برگ گفته می شود.
- **سطح گره:** ریشه معمولاً سطح صفر است (یک هم در نظر گرفته می شود). سطح بقیه گره ها، یک واحد بیشتر از سطح گره بالاتر آن است.
- **سطح درخت:** به بزرگترین سطح بین گره های برگ گفته می شود.
- **ارتفاع درخت:** حداکثر تعداد گره های موجود در مسیر ریشه تا برگ ها را ارتفاع درخت گویند که معمولاً با h نمایش داده می شود.
- **درخت یگانه:** درختی است که فقط دارای گره ریشه است.

سطوح گره‌ها
سطح ۰

عمق درخت = ۳
سطح درخت = ۳
ارتفاع درخت = ۴

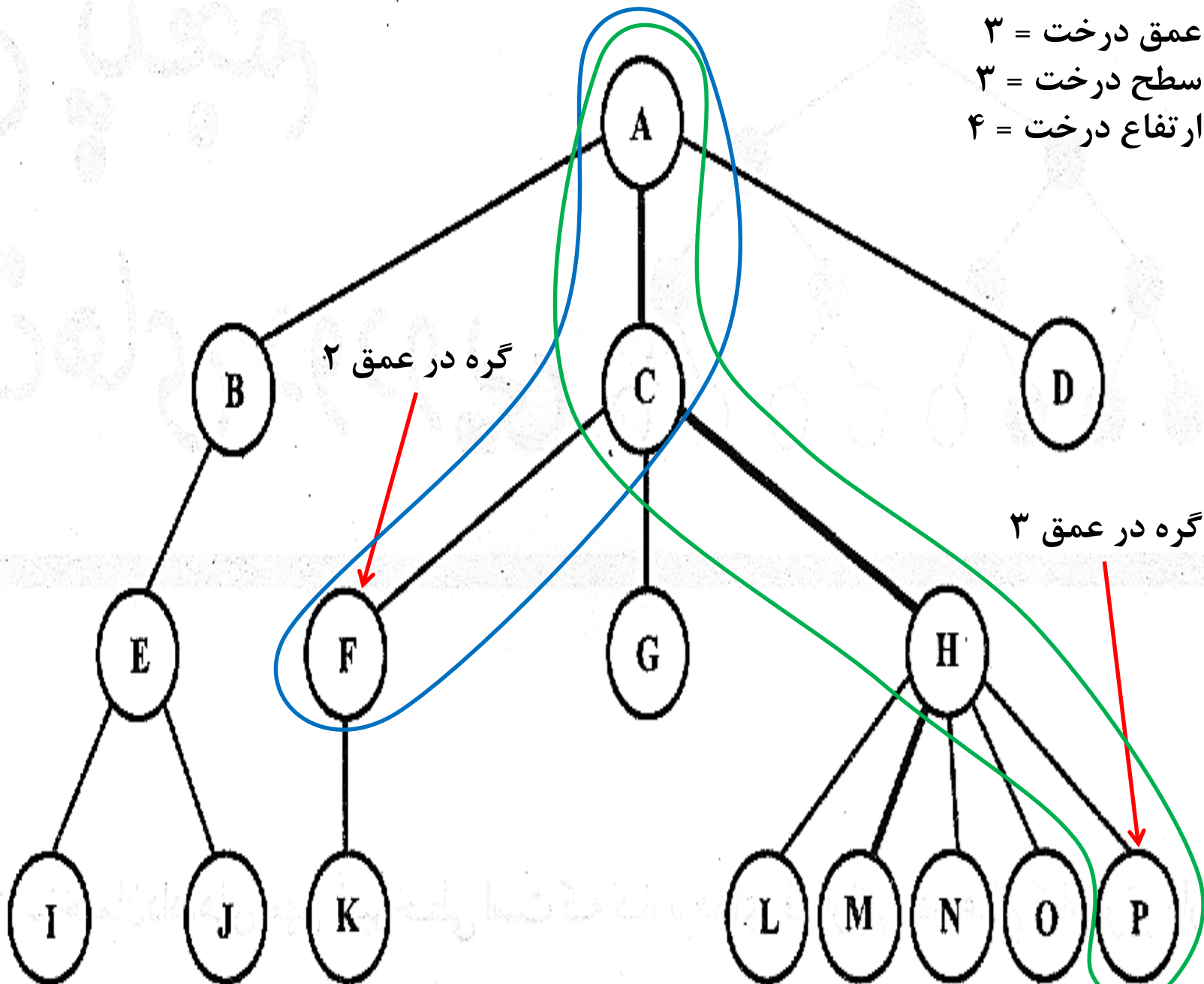
سطح ۱

گره در عمق ۲

سطح ۲

گره در عمق ۳

سطح ۳

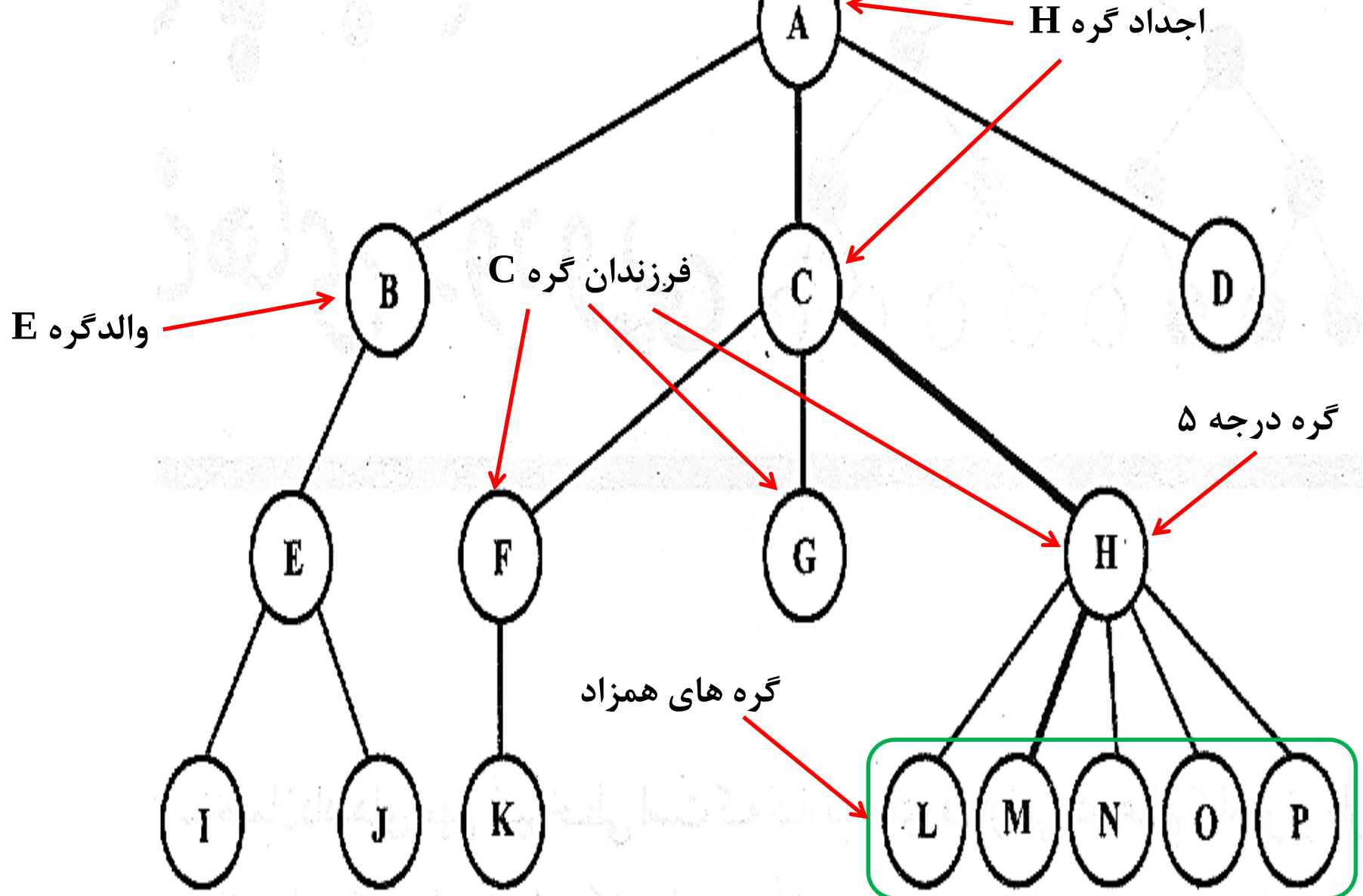


اصطلاحات مربوط به درخت ها

- **درخت خالی:** درختی که هیچ گره ای ندارد.
- **اجداد گره:** تمام گره های موجود در مسیر ریشه تا گره مورد نظر به جز خود آن گره را اجداد آن گره می گویند.
- **والد گره (پدر گره):** به جد بلافصل هر گره گفته می شود.
- **فرزندان گره:** به گره هایی گفته می شود که جد آنها گره مورد نظر است.
- **همزاد:** گره هایی که دارای جد بلافصل مشترک هستند، همزاد نام دارند.
- **اندازه درخت:** تعداد گره های موجود در درخت را اندازه درخت می گویند.
- **درجه گره:** تعداد فرزندان یک گره را درجه آن گره می گویند.

ریشه

اندازه درخت = ۱۶



مفهوم درخت دودویی

- درخت دودویی مجموعه محدودی از گره ها است که:

- a. یا تهی است.

- b. یا حاوی گره خاصی به نام ریشه است و بقیه گره های آن، دو زیر درخت دودویی مجزا به نام زیر درخت چپ و زیر درخت راست را تشکیل می دهند.

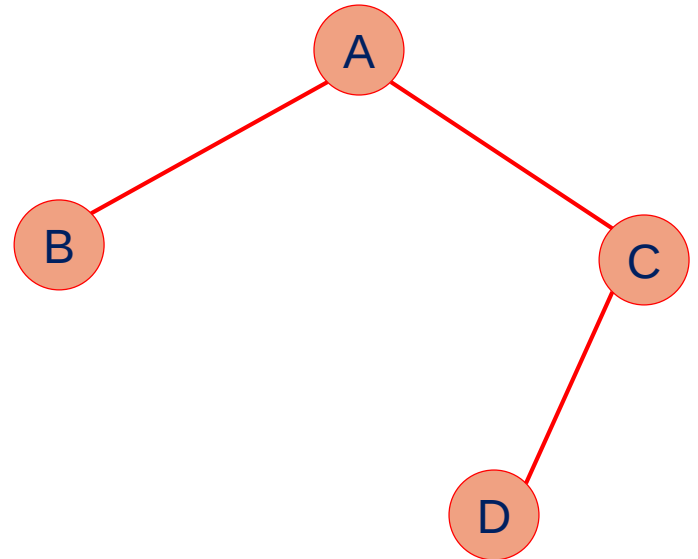
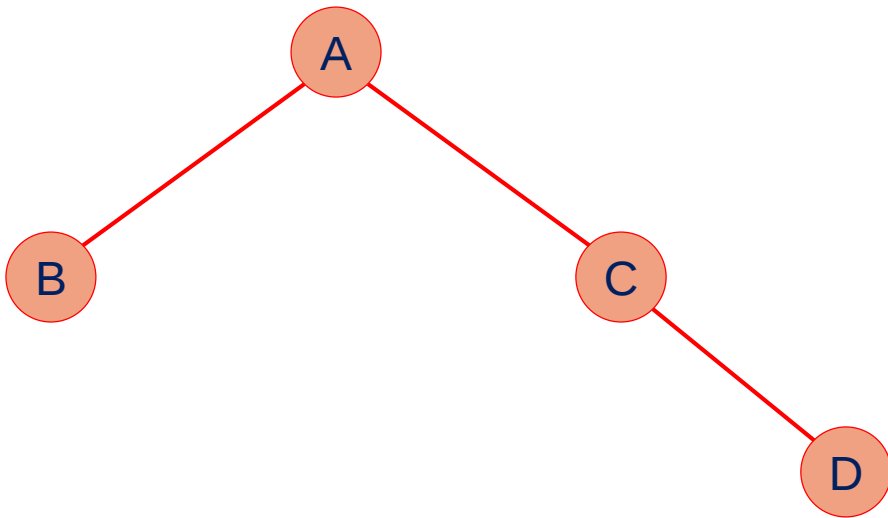
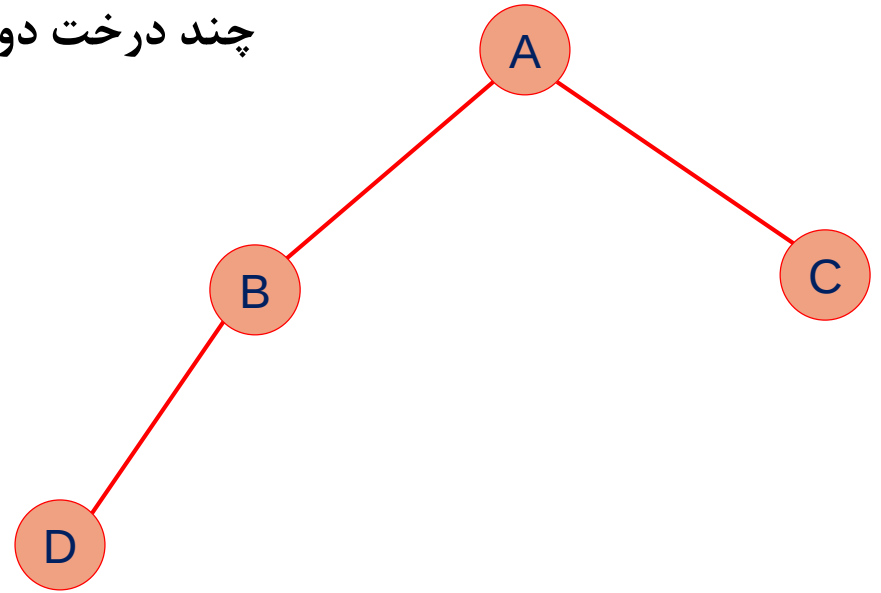
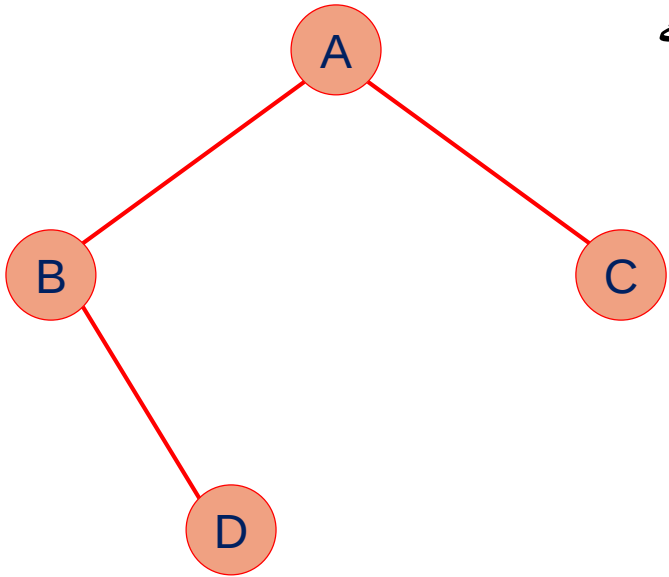
- تفاوت های بین درخت دودویی و درخت عمومی

- درخت دودویی می تواند تهی باشد ولی درخت عمومی نمی تواند.

- در درخت دودویی هر گره حداکثر دو فرزند دارد.

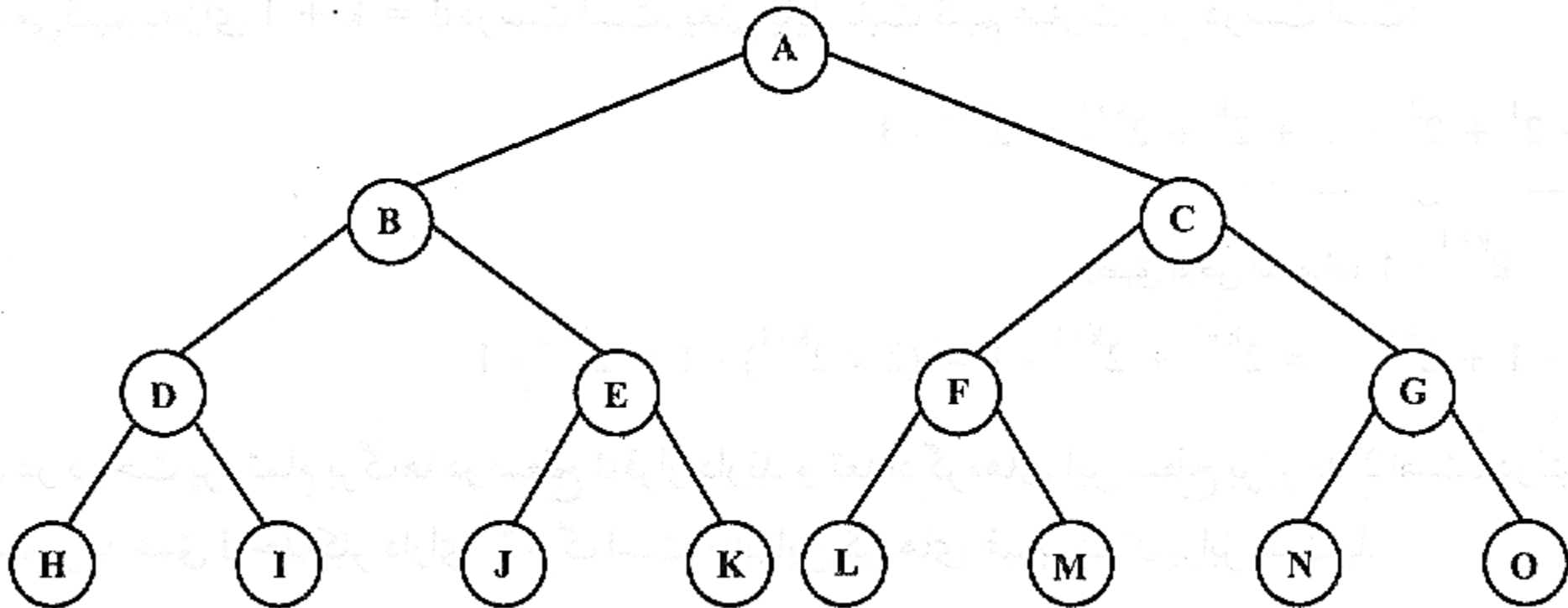
- ترتیب فرزندان در درخت دودویی اهمیت دارد در حالیکه در درخت عمومی اینطور نیست.

چند درخت دودویی نمونه



درخت دودویی پُر

- **درخت دودویی پُر (Full)** درختی است که تمام برگ های آن در یک سطح باشند و هر گره داخلی نیز دارای دو فرزند باشد.



درخت دودویی پُر

- درخت پُر در هر سطح i دقیقاً دارای 2^i گره است.
- اگر عمق درخت دودویی پُر را d و تعداد گره های آنرا tn بنامیم، آنگاه تعداد گره های درخت دودویی از رابطه زیر محاسبه می شود:

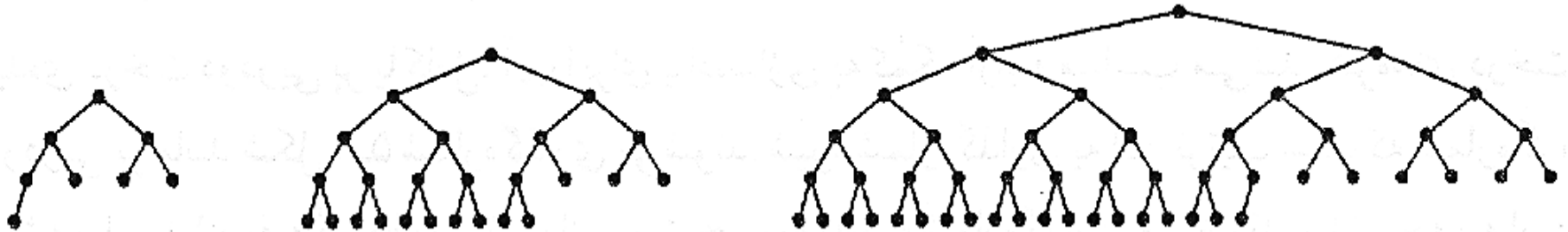
$$tn = 2^0 + 2^1 + 2^2 + \dots + 2^d = \sum_{j=0}^d 2^j = 2^{d+1} - 1$$

- اگر تعداد گره های درخت دودویی پُر مشخص باشد، آنگاه عمق درخت دودویی از رابطه زیر محاسبه می شود:

$$tn = 2^{d+1} - 1 \Rightarrow tn + 1 = 2^{d+1} \Rightarrow d = \log_2^{(tn+1)} - 1$$

درخت دودویی کامل

- **درخت دودویی کامل** درختی است که یا پُر است یا با افزودن گره‌های پشت سر هم به سمت راست سطح آخر درخت، به درخت پُر تبدیل خواهد شد.

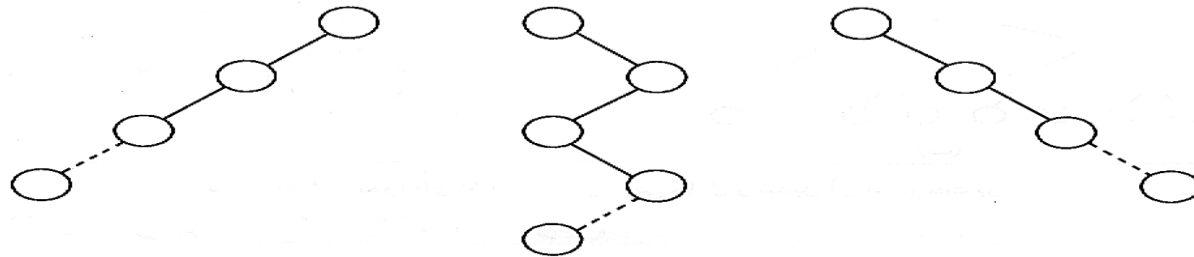


- اگر عمق درخت دودویی d باشد، برگ‌های این درخت در سطح d یا $d-1$ قرار دارند. اگر تعداد گره‌ها n باشد آنگاه:

$$d + 1 \leq n \leq 2^{d+1} - 1 \quad d = \lfloor \log n \rfloor$$

خواص درخت دودویی

- در هر درخت دودویی، حداکثر تعداد گره ها در سطح k برابر با 2^k است بطوریکه $k \geq 0$.
- حداکثر تعداد گره های یک درخت دودویی به ارتفاع h برابر با $2^h - 1$ است.
- حداقل تعداد گره های یک درخت دودویی به ارتفاع h برابر h است.



- در هر درخت دودویی غیر تهی، اگر n گره و e یال (پیوند) وجود داشته باشد، آنگاه $n = e + 1$ خواهد بود.

پیاده سازی درخت دودویی

- درخت های دودویی را به دو روش می توان نمایش داد:

۱. با استفاده از آرایه

۲. با استفاده از اشاره گر ها (نمایش پیوندی پویا)

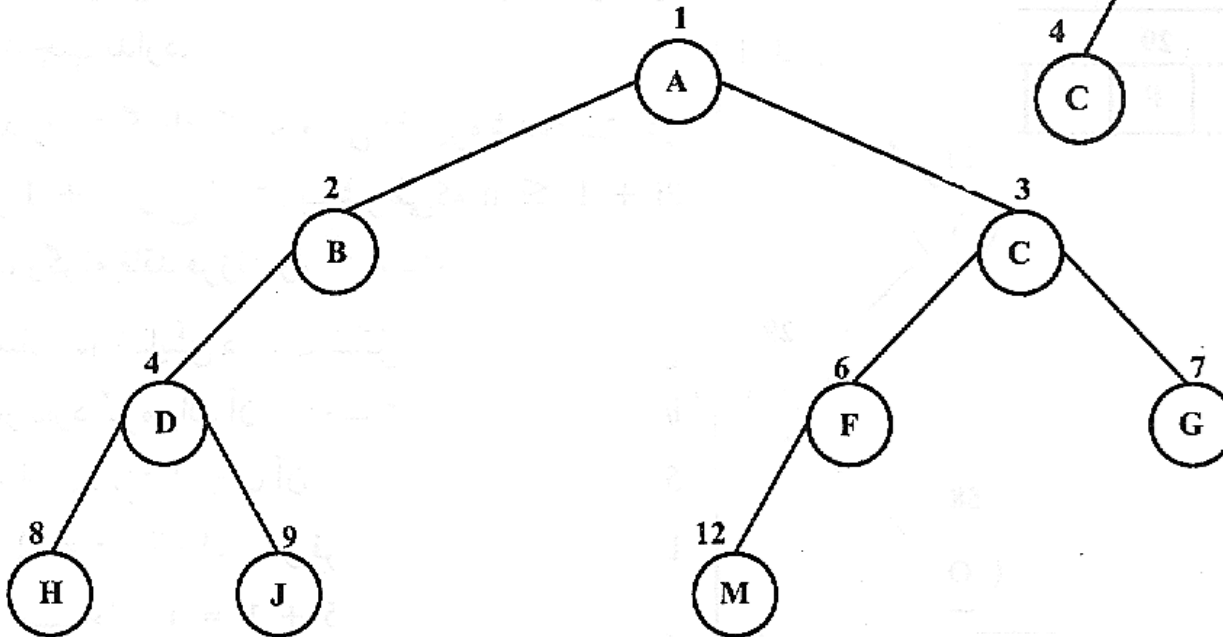
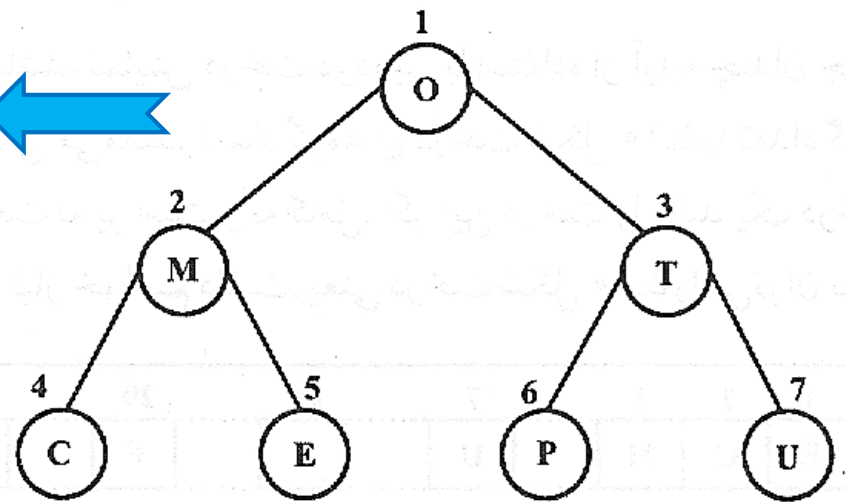
- آرایه برای پیاده سازی درخت های دودویی پُر یا کامل مناسب است و سایر درخت ها بهتر است با اشاره گر ها پیاده سازی شوند.

پیاده سازی درخت دودویی با آرایه

- (۱) فرض کنید درخت کامل است.
- (۲) به گره ریشه شماره 1 بدهید.
- (۳) سایر گره ها را سطح به سطح از سمت چپ به راست به ترتیب از شماره 2 به بعد قرار دهید.
- (۴) اگر درخت دارای n گره بود، در زبان C یک آرایه به اندازه $n+1$ تعریف می کنیم.
- (۵) گره های درخت را در سلول های آرایه با اندیس برابر با شماره آنها قرار می دهیم. سلول با اندیس 0 همیشه خالی خواهد ماند.

int tree1 [8];

0	1	2	3	4	5	6	7
	O	M	T	C	E	P	U

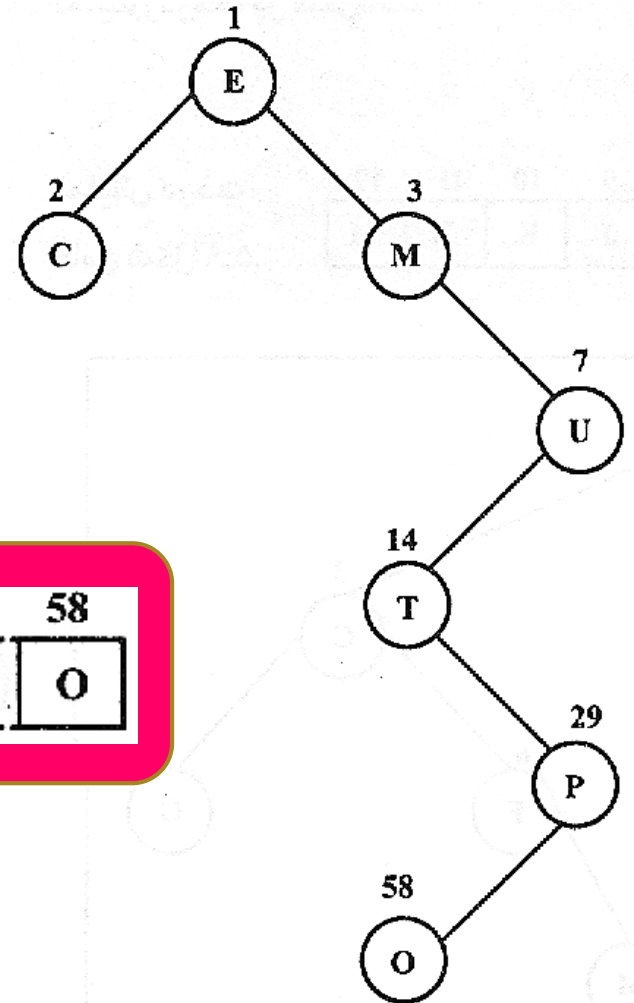


int tree2 [13];

0	1	2	3	4	5	6	7	8	9	10	11	12
	A	B	C	D		F	G	H	J			M

پیاده سازی درخت دودویی با آرایه

- اگر درخت دودویی پُر یا کامل نباشد، نمایش آن با آرایه چندان جالب نیست، زیرا عناصر زیادی از آرایه آن خالی می ماند.



0	1	2	3	7	14	29	58				
	E	C	M		U		T		P		O

ساخت درخت دودویی از آرایه

• دسترسی به گره های درخت دودویی در نمایش با آرایه بصورت زیر است:

(a) ریشه درخت در اندیس شماره 1 آرایه قرار دارد.

(b) پدر گره ای که در مکان i ذخیره شده است با شرط $i \neq 1$ ، در محل $i/2$ قرار دارد.

(c) فرزند چپ گره ای که در مکان i ذخیره شده است در محل $2i$ قرار دارد، به شرط اینکه $2i \leq n$ باشد.

(d) فرزند راست گره ای که در مکان i ذخیره شده است در محل $2i+1$ قرار دارد، به شرط اینکه $2i+1 \leq n$ باشد.

مزایا و معایب نمایش با آرایه

• مزایا

- ☆ هر گره از طریق گره دیگر قابل دسترسی است. (از طریق محاسبه اندیس)
- ☆ فقط داده ها ذخیره می شوند و نیاز به ذخیره سازی اشاره گرها نیست.
- ☆ در زبان هایی که تخصیص حافظه پویا ندارند، آرایه تنها روش نمایش است.

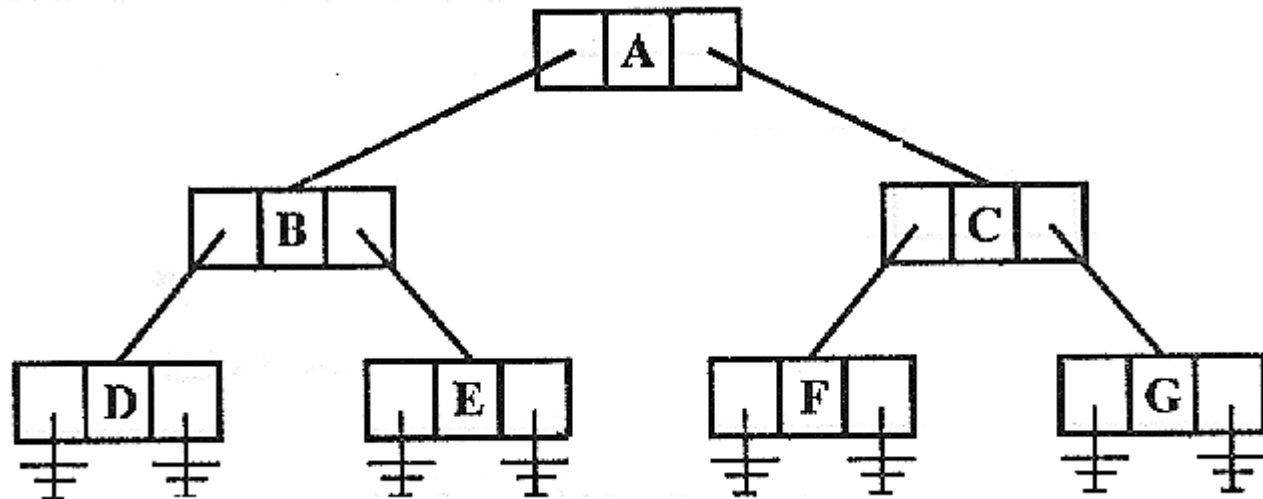
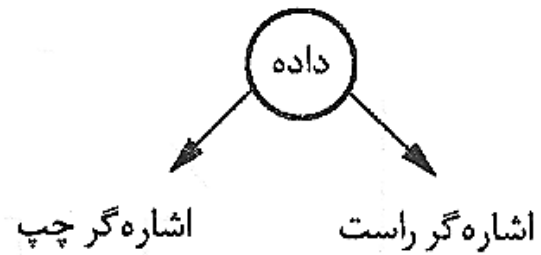
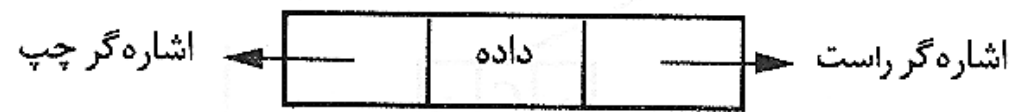
• معایب

- ☆ برای درخت های غیر پُر و غیر کامل، عناصر زیادی از آرایه خالی می مانند.
- ☆ با افزایش گره های درخت، طول آرایه قابل افزایش نیست.
- ☆ اعمال درج و حذف کارآمد نیست، زیرا نیاز به جابجایی عناصر آرایه دارد.

پیاده سازی درخت دودویی با اشاره گر

- برای نمایش درخت دودویی با اشاره گر، از لیست های پیوندی کمک گرفته می شود.
- هر گره در نمایش پیوندی دارای بخش داده، بخش اشاره گر راست و بخش اشاره گر چپ می باشد که اشاره گر راست به فرزند راست و اشاره گر چپ به فرزند چپ اشاره می کند.

```
struct  nodeType {  
    nodeType *left;  
    int info;  
    nodeType *right;  
};  
nodeType *node;
```



پیمایش درخت دودویی

- منظور از **پیمایش درخت دودویی**، حرکت در سراسر درخت و ملاقات تمام گره های آن می باشد بطوریکه هر گره فقط یکبار ملاقات شود.
- سه روش متداول برای پیمایش درخت ها وجود دارد:
 - ➊ **روش پیشوندی (Preorder)**
 - ➋ **روش میانوندی (Inorder)**
 - ➌ **روش پسوندی (Postorder)**
- هریک از این سه روش را می توان بصورت بازگشتی و غیر بازگشتی طراحی کرد که در عمل پیاده سازی بازگشتی آنها ساده تر می باشد.

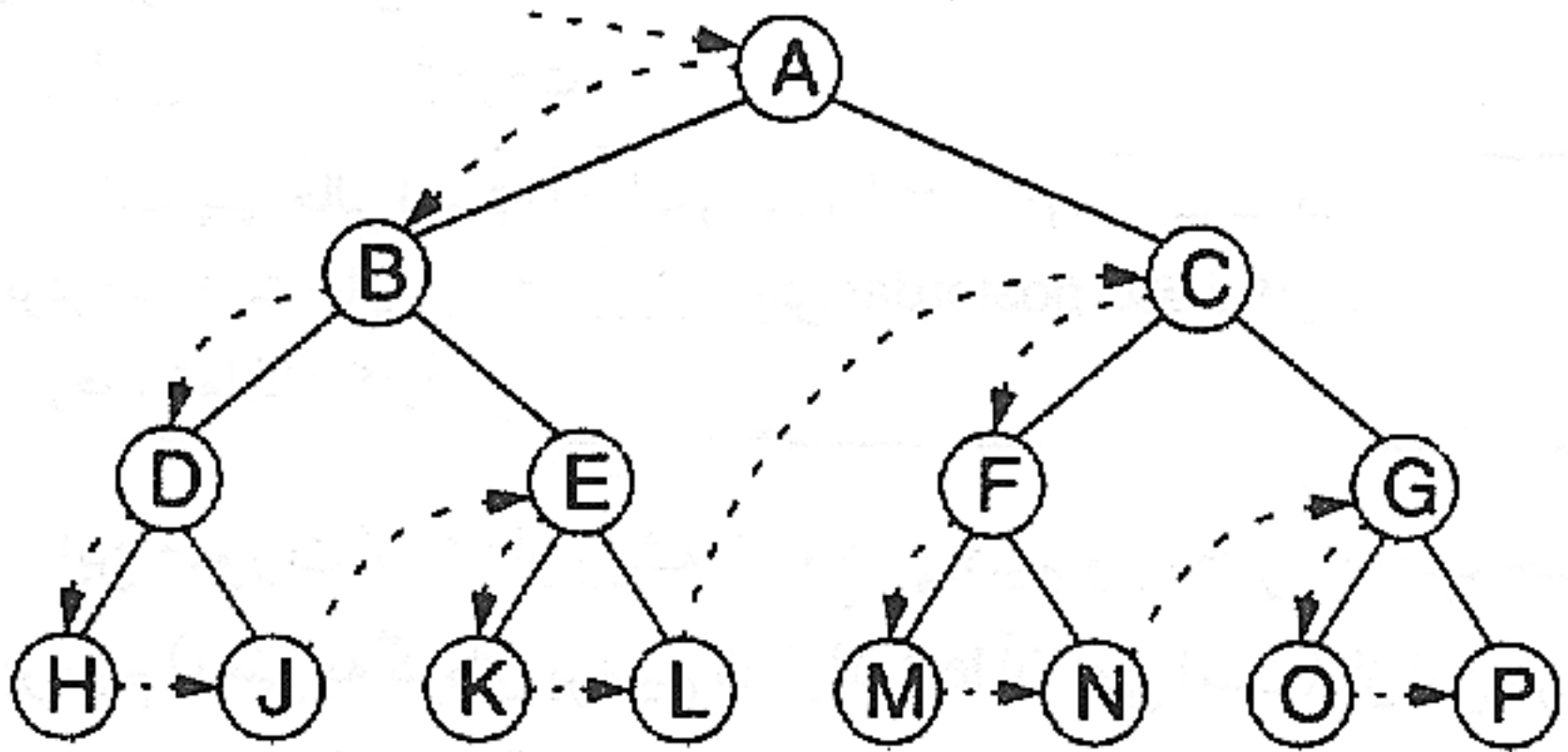
روش پیمایش پیشوندی

(۱) ریشه درخت را ملاقات کنید.

(۲) اگر زیر درخت چپ خالی نیست، آنرا به روش پیشوندی پیمایش کنید.

(۳) اگر زیر درخت راست خالی نیست، آنرا به روش پیشوندی پیمایش کنید.

```
void preorder(nodeType *node)
{
    if(node)
    {
        printf("%d", node -> info);
        preorder(node -> left);
        preorder(node -> right);
    }
}
```

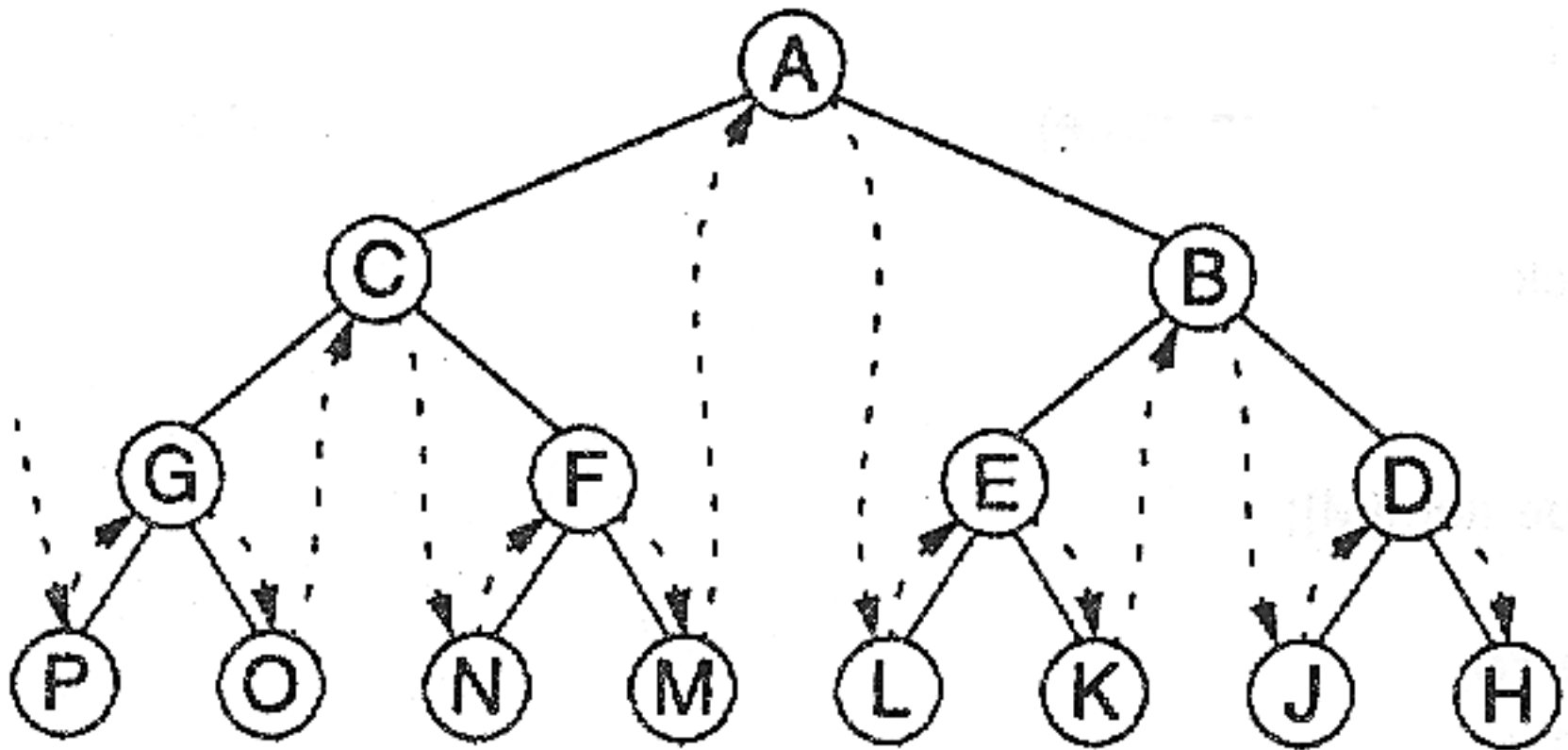


ABDHJEKLCFMNGOP

روش پیمایش میانوندی

- (۱) اگر زیر درخت چپ خالی نیست، آنرا به روش میانوندی پیمایش کنید.
- (۲) ریشه درخت را ملاقات کنید.
- (۳) اگر زیر درخت راست خالی نیست، آنرا به روش میانوندی پیمایش کنید.

```
void inorder(nodeType *node)
{
    if(node)
    {
        inorder(node -> left);
        printf("%d", node -> info);
        inorder(node -> right);
    }
}
```

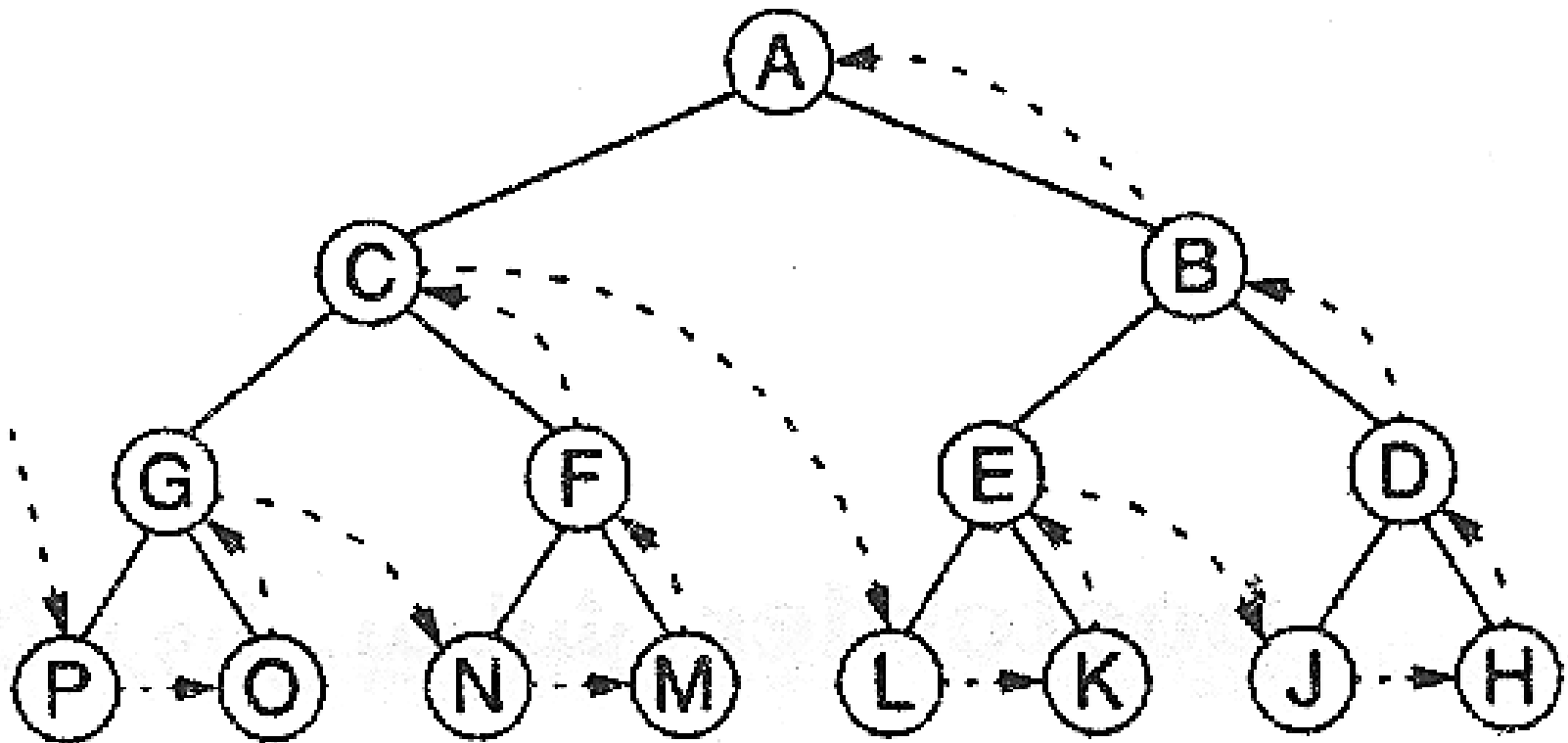


PGOCNFMAL E K B J D H

روش پیمایش پسوندی

- (۱) اگر زیر درخت چپ خالی نیست، آنرا به روش پسوندی پیمایش کنید.
- (۲) اگر زیر درخت راست خالی نیست، آنرا به روش پسوندی پیمایش کنید.
- (۳) ریشه درخت را ملاقات کنید.

```
void postorder(nodeType *node)
{
    if(node)
    {
        postorder(node -> left);
        postorder(node -> right);
        printf("%d", node -> info);
    }
}
```



POGNMFLKEJHDBA

ساخت درخت دودویی از روی پیمایش

- با استفاده از ترتیب گره های حاصل از پیمایش یک درخت دودویی، شکل واقعی خود درخت را می توان بدست آورد.
- اگر یک پیمایش درخت موجود باشد، نمی توان درخت منحصر به فرد مورد نظر را ایجاد کرد.
- اگر پیمایش های پیشوندی و پسوندی یک درخت موجود باشد، نمی توان درخت منحصر به فرد مورد نظر را ایجاد کرد.
- اگر پیمایش میانوندی و یکی از پیمایش های پیشوندی یا پسوندی موجود باشد، می توان درخت منحصر به فرد مورد نظر را ایجاد کرد.

ساخت درخت دودویی از روی پیمایش

- قواعد اصلی برای ساخت درخت دودویی با استفاده از پیمایش های آن بصورت زیر است:

* اگر پیمایش پیشوندی درخت دودویی مشخص باشد، اولین گره لیست، ریشه درخت دودویی است.

* اگر پیمایش پسوندی درخت دودویی مشخص باشد، آخرین گره لیست، ریشه درخت دودویی است.

* وقتی گره ریشه مشخص شد، به کمک پیمایش میانوندی درخت می توان زیر درخت چپ و زیر درخت راست را پیدا کرد.

مثال : با توجه به پیمایش های میانوندی و پیشوندی درخت دودویی، خود درخت را ترسیم کنید.

Inorder :	D	B	H	E	A	I	F	J	C	G
Preorder:	A	B	D	E	H	C	F	I	J	G

مثال : با توجه به پیمایش های میانوندی و پسوندی درخت دودویی، خود درخت را ترسیم کنید.

Inorder :	N1	N2	N3	N4	N5	N6	N7	N8	N9
Postorder:	N1	N3	N5	N4	N2	N8	N7	N9	N6

