

به نام خدا

# ساختمان داده ها

جلسه دوازدهم

دانشگاه صنعتی همدان

گروه مهندسی کامپیوتر

نیم سال دوم 1397-98

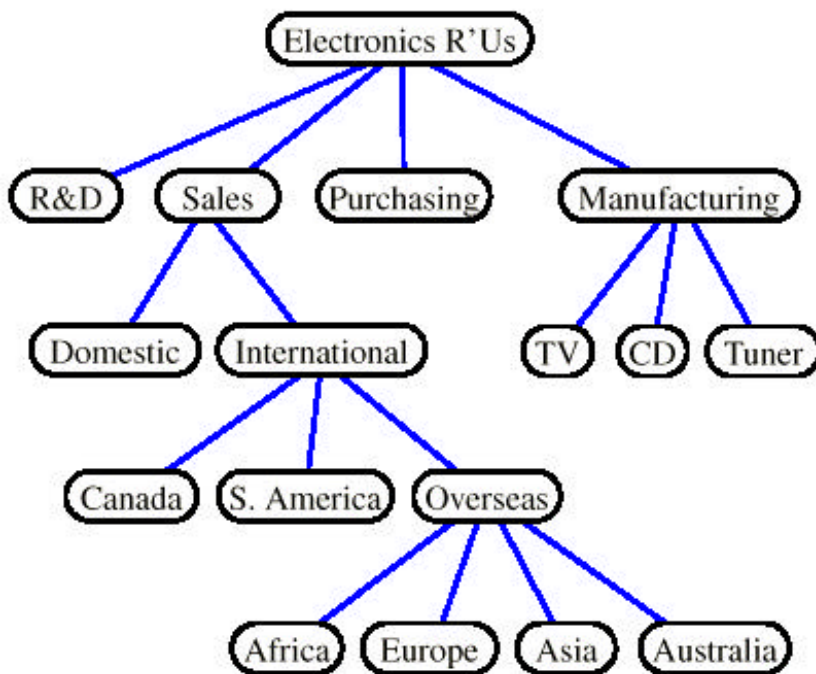
# درختان

# Trees

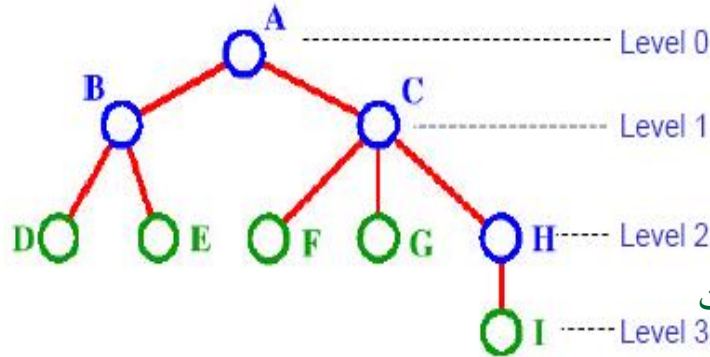
- مقدمه و تعاریف
- نمایش درختان
- درختان دودویی
  - ADT درختان دودویی
  - خواص درختان دودویی
  - نمایش درختان دودویی
  - نمایش درختان با لیست ها
  - پیمایش درختان دودویی
  - اعمال روی درختان دودویی
  - درختان دودویی نخ کشی شده
- هرمها (Heaps)
- درختان جستجو
- درختان انتخاب
- جنگلها

## مقدمه و تعاریف

- ساختار درختی مجموعه ای از داده های سازماندهی شده که عناصر اطلاعاتی با انشعاباتی به یکدیگر متصل شده اند.
- مانند نمودار وراثت ؛ شجره نامه و نمودار سازمانی (مثال زیر)



## تعریف درخت



■ درخت مجموعه ای متناهی از یک یا چند گره

■ است به طوری که:

○ دارای یک گره ریشه باشد.

○ بقیه گره ها به  $n \geq 0$  مجموعه مجزا

○  $T_1 \dots T_n$  تقسیم شده که خود این مجموعه ها درخت

○ هستند. به این مجموعه ها زیر درختان ریشه می گویند.

## ■ چند اصطلاح

○ درجه گره - تعداد زیر درختان یک گره

○ برگ - هر گره با درجه صفر

○ والد و فرزند

○ همزاد - گره های که والد یکسان دارند

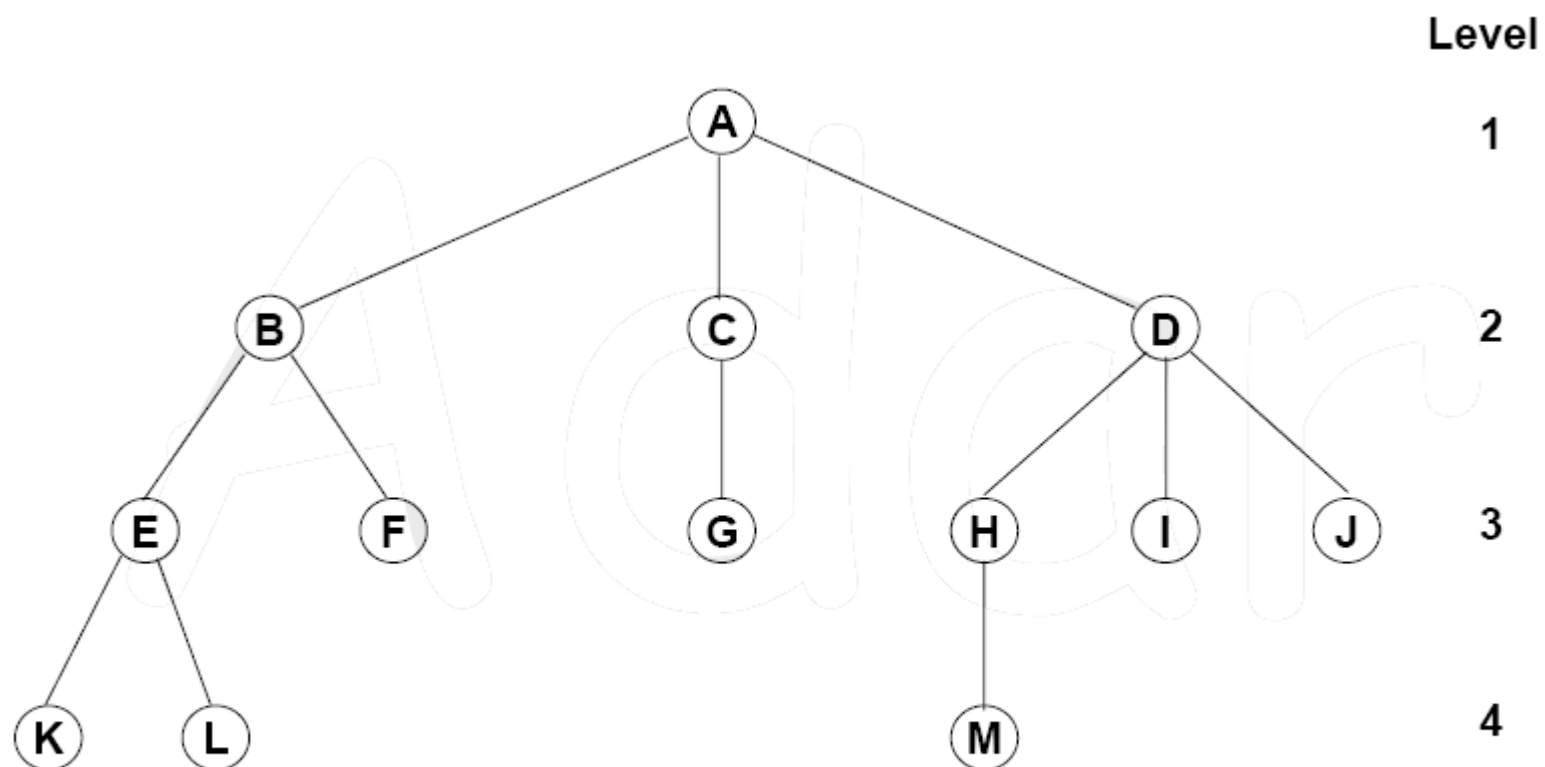
○ درجه درخت - بزرگترین درجه گره ها

○ اجداد گره - گره هایی که در مسیر ریشه یک گره هستند

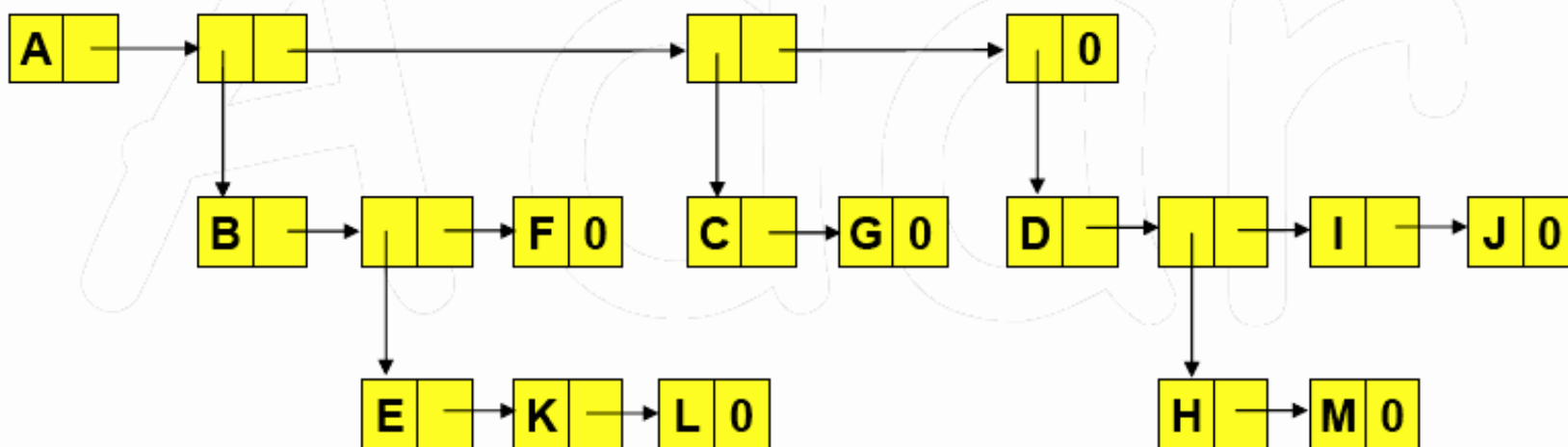
○ سطح - اگر گره ریشه در سطح یک باشد هر گره که در سطح  $L$  باشد فرزندان

آن در سطح  $L+1$  هستند.

○ ارتفاع یا عمق درخت - تعداد سطوح یک درخت



**(A(B(E(K,L),F),C(G),D(H(M),I,J)))**



- روش لیستی با اندازه گره ثابت

Data	Child 1	Child 2	Child 3	Child 4	...	Child k
------	---------	---------	---------	---------	-----	---------

- در این روش حافظه زیادی هدر می رود زیرا در یک درخت با  $n$  گره و درجه  $k$  تعداد  $nk - (n - 1)$  فیلد صفر یا نول وجود دارد.

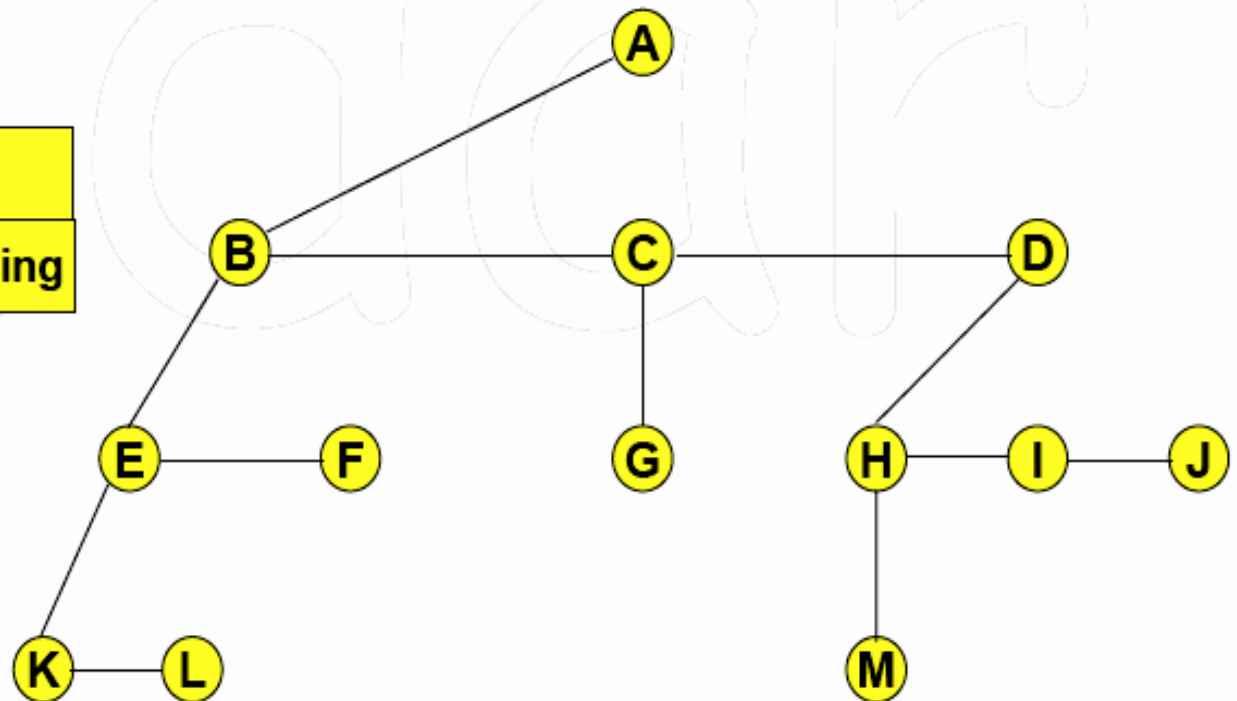


## نمایش درختان

• روش فرزند چپ همزاد راست

■ در این روش از گره ریشه شروع می کنیم و فرزند چپ آخر آن را به عنوان فرزند چپ در نظر می گیریم و همزاد راست آنرا به عنوان فرزند راست آن

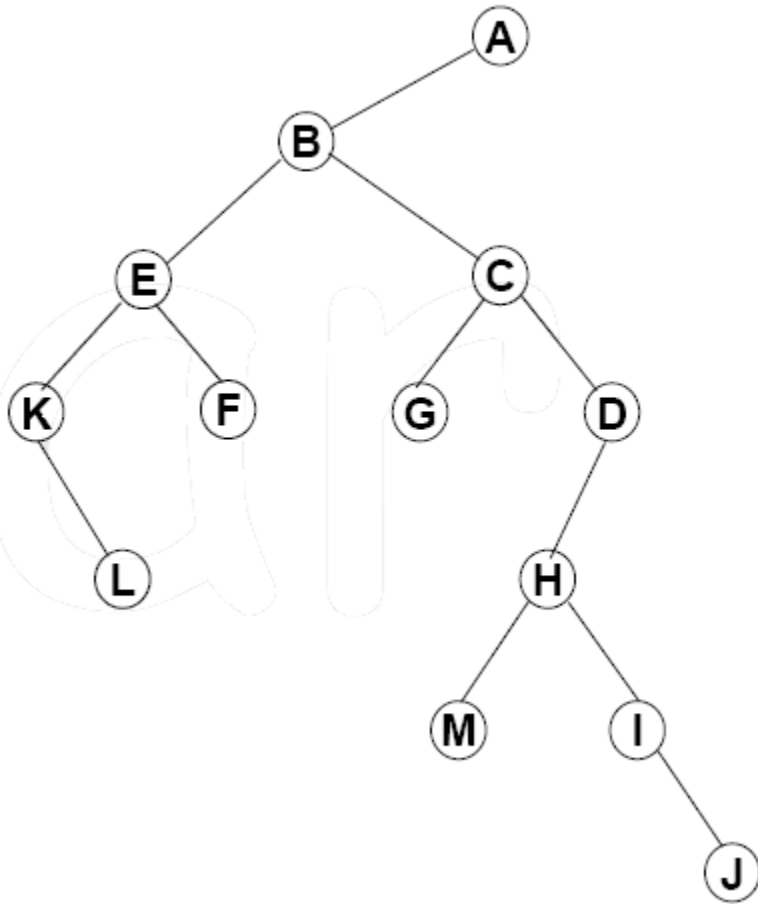
data	
left child	right sibling



# نمایش درختان

## • نمایش دودویی

- اگر یک نمایش فرزند چپ همزاد راست را به میزان ۴۵ درجه بچرخانیم یک درخت دودی داریم.

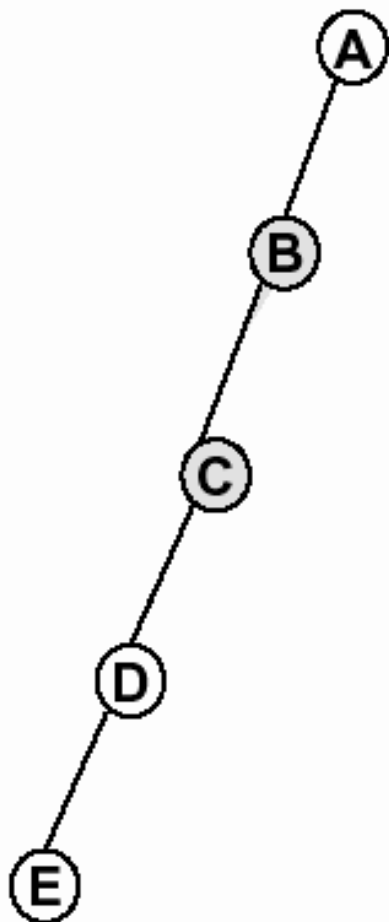


### • درخت دودویی:

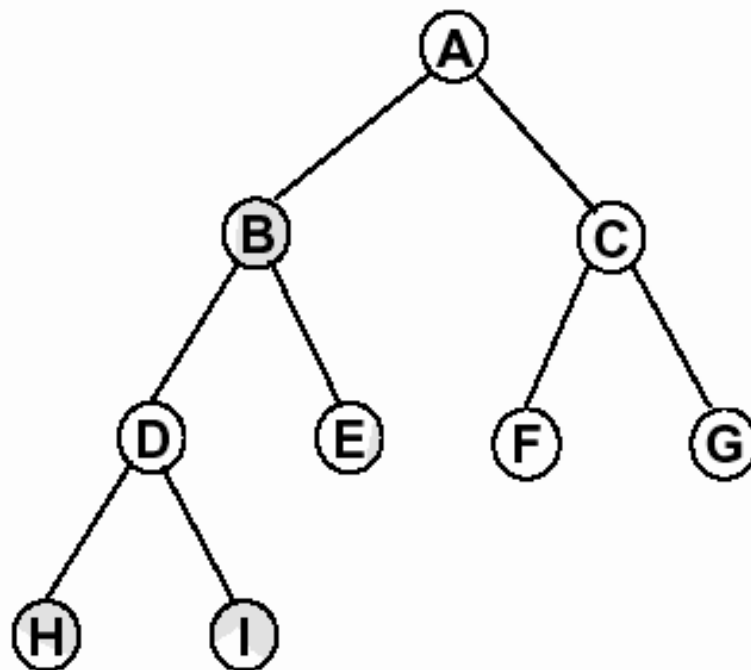
- در این درخت درجه هر نود حداکثر دو است.
- ترتیب نودها اهمیت دارد.
- ممکن است دارای صفر نود باشد.

### • تعریف رسمی:

- یک درخت دودویی مجموعه محدودی از نودها است که یا خالی است یا شامل ریشه و دو زیر درخت متمایز دودویی است که به آنها زیر درختان سمت راست و سمت چپ گفته می شود.



خطی - نامتعادل



کامل — متعادل تر

- خواص جالب درخت دودویی:

- حداکثر تعداد گره ها در سطح  $i$  برابر  $2^{i-1}$  است.

- حداکثر تعداد گره ها در درخت باینری با ارتفاع  $k$  ، برابر  $2^k - 1$  است.

- برای هر درخت غیر تهی مانند  $T$  اگر  $n_0$  تعداد گره های درجه صفر و  $n_2$  تعداد گره های درجه ۲ باشد آنگاه  $n_0 = n_2 + 1$

- یک درخت باینری پر با عمق  $K$  ، دارای  $2^k - 1$  نود هست

## درختان دودیی

■ حداکثر تعداد نودها در سطح  $i$  برابر  $2^{i-1}$  است.

● اثبات با استفاده از استقراء:

■ حالت پایه:

$$\text{Level } 1 = 2^{1-1} = 2^0 = 1 \quad \circ$$

■ برای یک  $i > 1$

○ اگر سطح  $i-1$  دارای  $2^{i-2}$  نود باشد.

■ برای  $i$ :

○ هر نود در سطح  $i-1$ ، حداکثر دو فرزند در سطح  $i$  دارد. پس تعداد نودهای سطح  $i$  حداکثر برابر  $2^{i-1} * 2 = 2^i$  است.

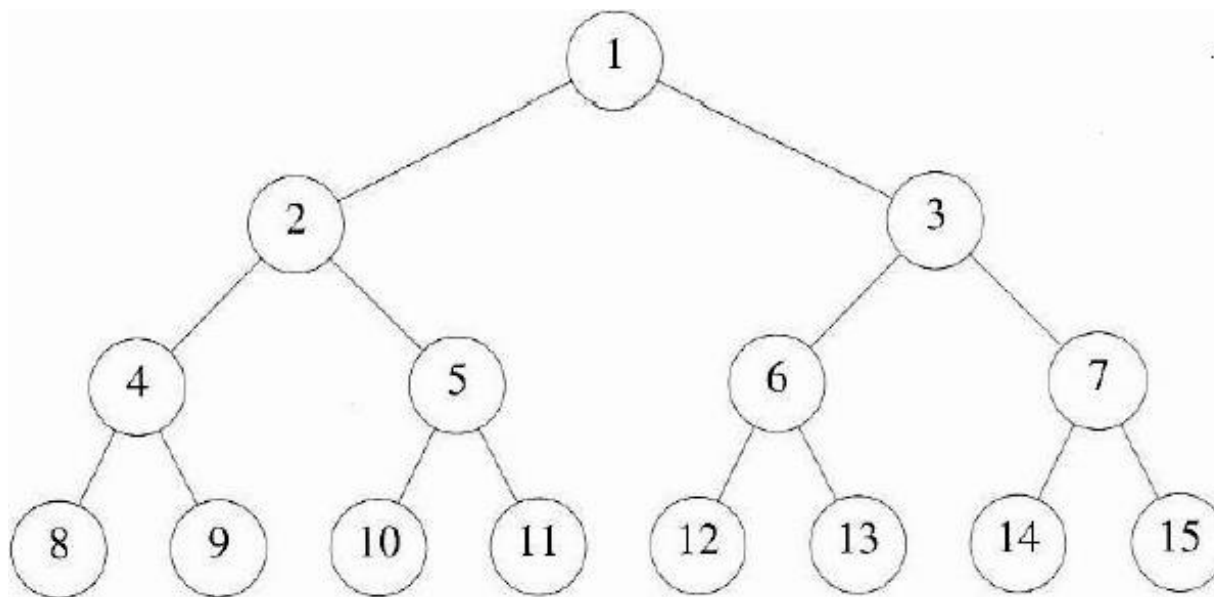
## درختان دودیی

- حداکثر تعداد نودها در درخت باینری با ارتفاع  $k$ ، برابر  $2^k - 1$  است. می توان ماکسیمم تعداد نودهای هر سطح را با هم جمع زد. باید حاصل جمع  $1 + 2 + 4 + \dots + 2^{k-1}$  را بدست آوریم. این جمع برابر است با:  $2^k - 1$

- برای هر درخت غیر تهی مانند  $T$  اگر  $n_0$  تعداد گره های درجه صفر و  $n_2$  تعداد گره های درجه ۲ باشد انگاه  $n_0 = n_2 + 1$

- $n = n_0 + n_1 + n_2$
- $n = B + 1$
- $B = n_1 + 2n_2$
- $n = B + 1 = n_1 + 2n_2 + 1$
- $n_0 + n_1 + n_2 = n_1 + 2n_2 + 1$

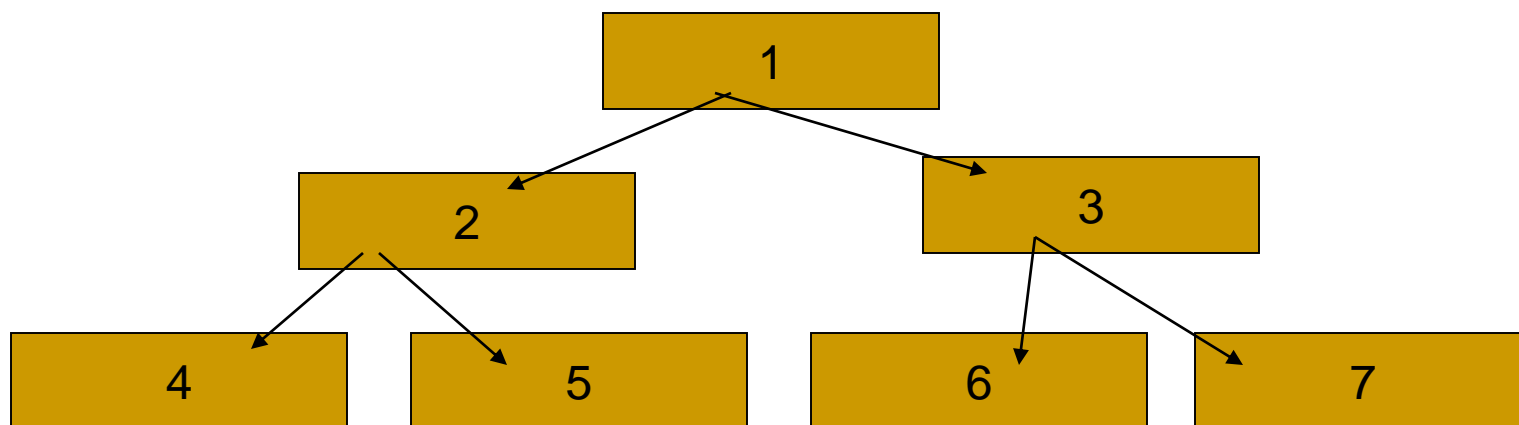
- یک درخت باینری پر با عمق  $K$ ، دارای  $2^k - 1$  نود هست
- یک درخت دودویی به عمق  $k$  کامل است اگر شماره گذاری گره های آن مطابق با شماره گذاری یک درخت دودویی پر به عمق  $k$  باشد.
- ارتفاع یک درخت دودویی کامل با  $n$  نود برابر است با:  $(\log_2(n)+1)$





## نمایش درختهای دودویی

- پیاده سازی آرایه ای



- هر نود با یک عنصر از آرایه متناظر می گردد

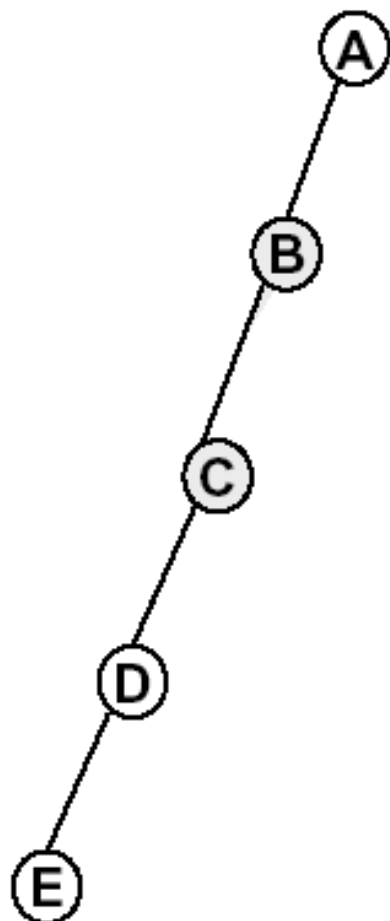
## نمایش آرایه ها با آرایه

• پیاده سازی آرایه ای:

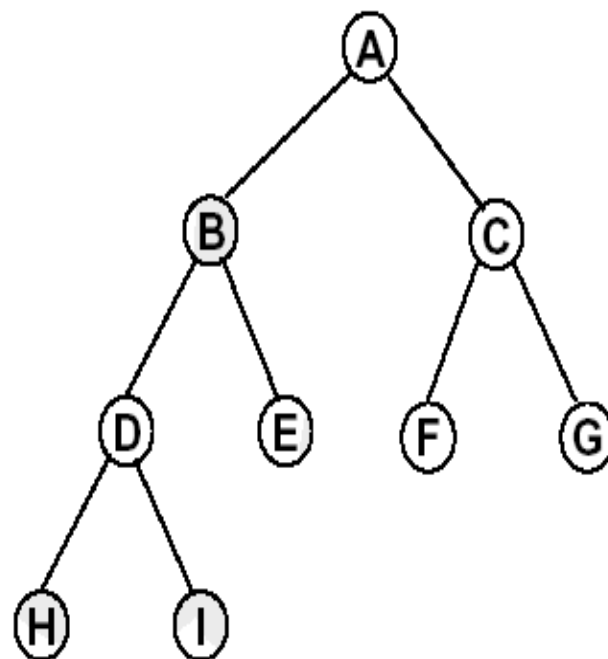
- $\text{parent}(i)$  در محل  $i/2$  قرار دارد (برای  $i \neq 1$ ).
- فرزند سمت چپ  $i$  در محل  $2i$  و فرزند سمت راست در محل  $2i+1$  قرار دارد.
- اگر  $2i > n$  باشد،  $i$  فرزند سمت چپ ندارد.
- اگر  $2i+1 > n$  باشد،  $i$  فرزند سمت راست ندارد.

Array Index: 0      1      2      3      4      5      6      7

خالی	Data1	Data2	Data3	Data4	Data5	Data6	Data7
------	-------	-------	-------	-------	-------	-------	-------



A
B
—
C
—
—
—
D
—
•
•
•
E



[1]  
[2]  
[3]  
[4]  
[5]  
[6]  
[7]  
[8]  
[9]

A
B
C
D
E
F
G
H
I

## نمایش درختها با آرایه

- نمایش آرایه ای بهترین راه حل نیست.

- اندازه حافظه ثابت است:

- براحتی قابل گسترش نیست.

- اگر درخت بالانس نباشد، حافظه هدر می رود.

- راه بهتر: استفاده از ایده لینک پیوندی.

- می شود از راه حل استفاده از همزاد دوری کرد، چون تعداد فرزندان ثابت است و فقط

- به دو اشاره گر نیازمندیم.