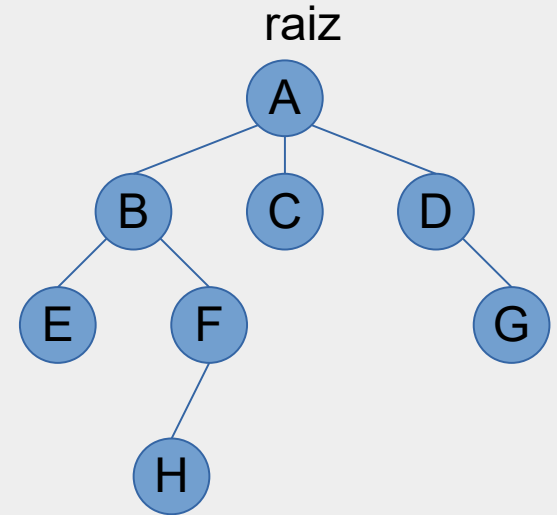


## GSI011 – Estrutura de Dados 2

Prof<sup>a</sup>: Christiane Regina Soares Brasil

# GSI011 – Estrutura de Dados 2 – Aula 10

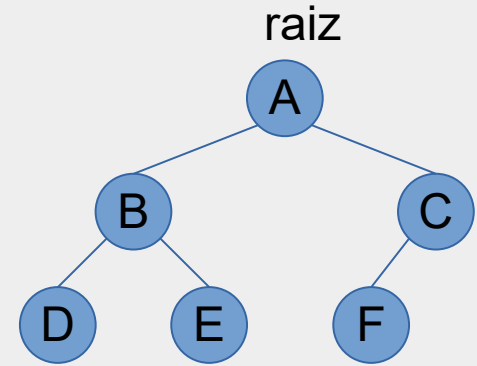
- Principais Conceitos de Árvores, de modo geral:
  - Raiz: nó que não possui pai.
  - Pai: nó antecessor direto de outro nó.
  - Filho: nó sucessor direto de outro nó.
  - Nó folha: aquele que não possui filhos.
  - Nó interno: aquele que possui, pelo menos, um filho e **não** é raiz.
  - Caminho: percurso de um nó **vi** até um nó **vf**.



3 nós internos,  
4 folhas e  
ABFH é um caminho.

# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Array (alocação estática).
  - Lista Encadeada (alocação dinâmica).



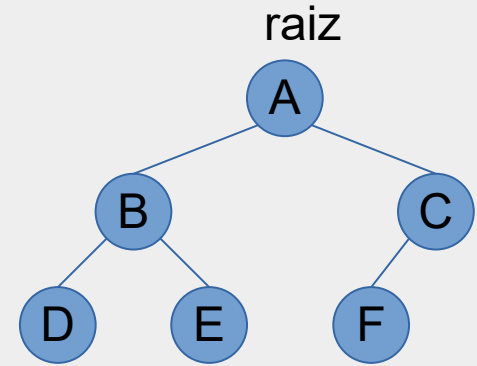
# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:

- Array (alocação estática).

- Filho Esq (Pai) =  $2 * \text{Pai} + 1$
    - Filho Dir (Pai) =  $2 * \text{Pai} + 2$

0	1	2	3	4	5	...
A	B	C	D	E	F	



raiz = posição 0, inicialmente.

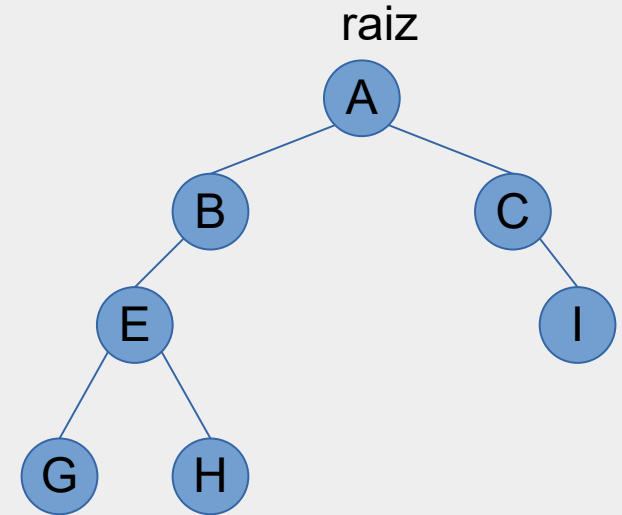
Filho esq(A) = B =  $2 * 0 + 1 = 1$   
Filho dir (A) = C =  $2 * 0 + 2 = 2$   
Filho esq(B) = D =  $2 * 1 + 1 = 3$   
Filho dir (B) = E =  $2 * 1 + 2 = 4$   
Filho esq(C) = F =  $2 * 2 + 1 = 5$

# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:

- Array (alocação estática).
  - Filho Esq (Pai) =  $2 * \text{Pai} + 1$
  - Filho Dir (Pai) =  $2 * \text{Pai} + 2$

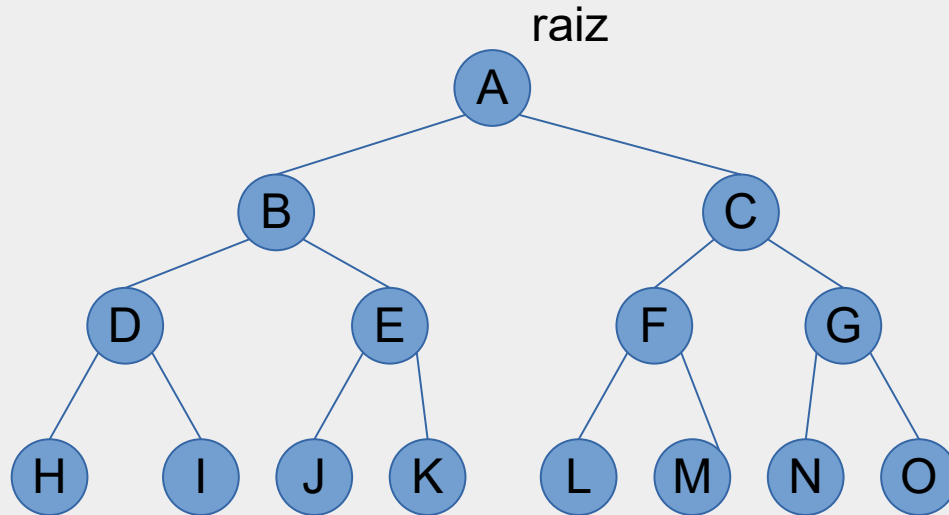
0	1	2	3	4	5	6	7	8	...
A	B	C	E	--	--	I	G	H	



# GSI011 – Estrutura de Dados 2 – Aula 10

## Tipos de Representação de Árvore Binária:

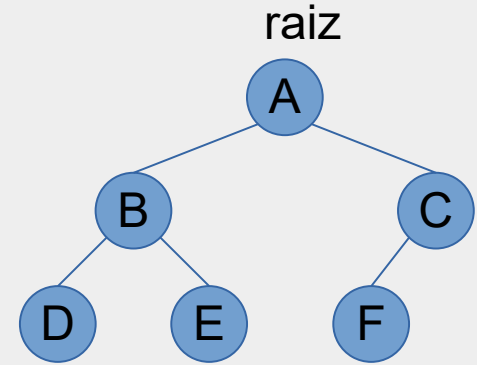
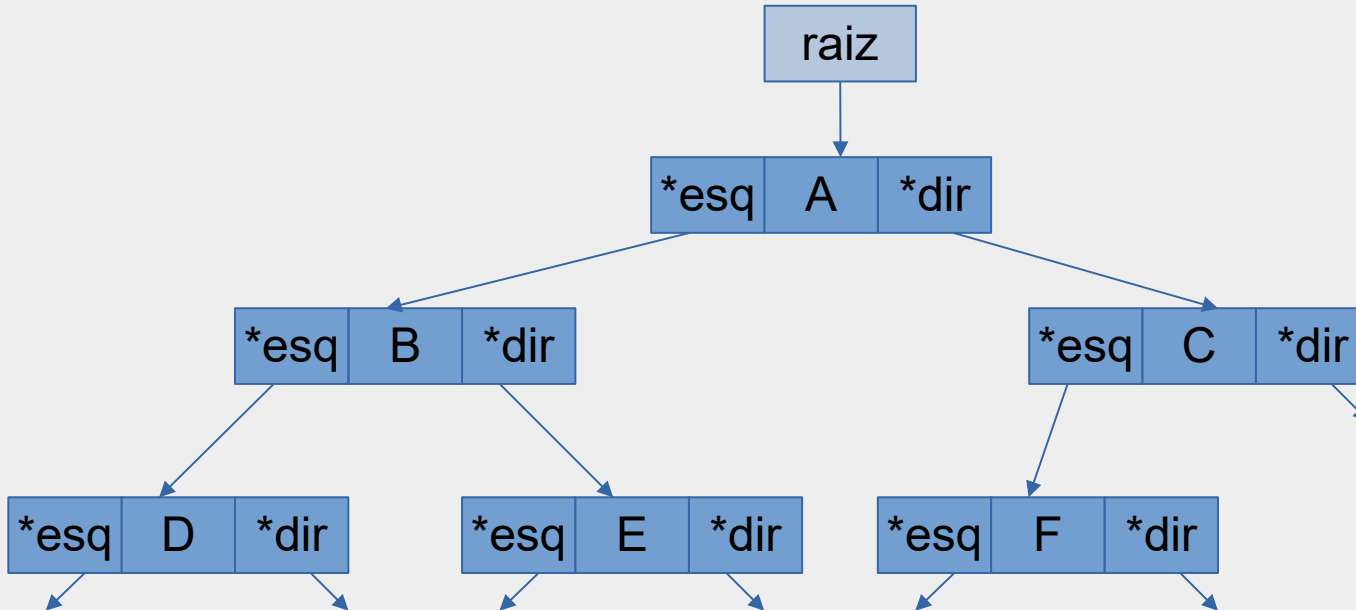
- Array (alocação estática).



Altura (h)	Máx. Nós
1	1
2	1 + 2
3	3 + 4
4	7 + 8
...	...
h	$2^h - 1$

# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Lista Encadeada (alocação dinâmica).



# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Lista Encadeada (alocação dinâmica).

```
//ArvoreBinaria.h  
  
typedef struct No* ArvBin;  
...  
  
//ArvoreBinaria.c  
#include "ArvoreBinaria.h"  
struct No{  
    int info;  
    struct No*esq;  
    struct No*dir;  
};
```

*esq	A	*dir
------	---	------

Capítulo 11, pág. 342



# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Lista Encadeada (alocação dinâmica).

```
//ArvoreBinaria.h
```

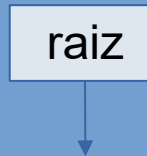
```
typedef struct No* ArvBin;  
ArvBin* criaArvBin( );  
void liberaArv(ArvBin* raiz);  
void liberaNo(struct No* no);
```

```
//ArvoreBinaria.c  
//Criando a árvore  
ArvBin * criaArvBin( )
```

```
{  
    ArvBin *raiz = (ArvBin*) malloc (sizeof(ArvBin));  
    if(raiz != NULL)  
        *raiz = NULL;  
    return raiz;  
}
```

```
//main.c
```

```
ArvBin *raiz = criaArvBin ( );
```



# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Lista Encadeada (alocação dinâmica).

```
//ArvoreBinaria.h
```

```
typedef struct No* ArvBin;  
ArvBin* criaArvBin( );  
void liberaArv(ArvBin* raiz);  
void liberaNo(struct No* no);
```

```
//ArvoreBinaria.c  
//Liberando a árvore
```

```
void liberaArv(ArvBin* raiz)  
{  
    if(raiz == NULL) return;  
    liberaNo(*raiz);  
    free(raiz);  
}
```

# GSI011 – Estrutura de Dados 2 – Aula 10

- Tipos de Representação de Árvore Binária:
  - Lista Encadeada (alocação dinâmica).

```
//ArvoreBinaria.h
```

```
typedef struct No* ArvBin;  
ArvBin* criaArvBin( );  
void liberaArv(ArvBin* raiz);  
void liberaNo(struct No* no);
```

```
//ArvoreBinaria.c  
//Liberando o nó
```

```
void liberaNo(struct No* no)  
{  
    if( no == NULL) return;  
    liberaNo(no->esq);  
    liberaNo(no->dir);  
    free(no);  
    no = NULL;  
}
```

