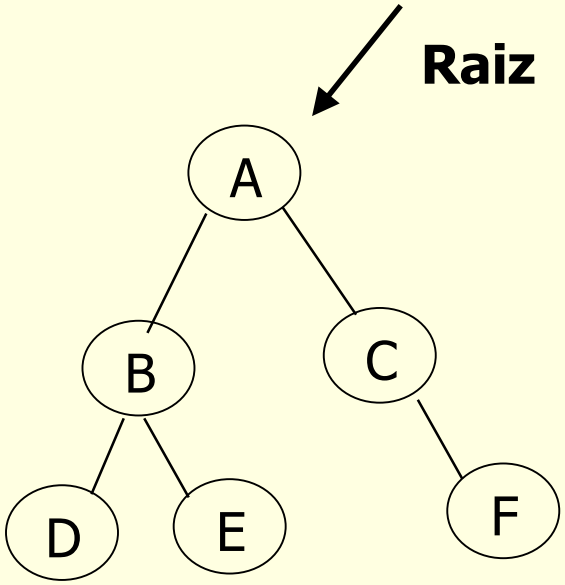
Árvores

SCC-202 – Algoritmos e Estruturas de Dados I

Prof. Thiago A. S. Pardo

Árvores binárias

- Árvores com grau 2, ou seja, cada nó pode ter 2 filhos, no máximo



Terminologia:

- filho esquerdo
- filho direito
- informação

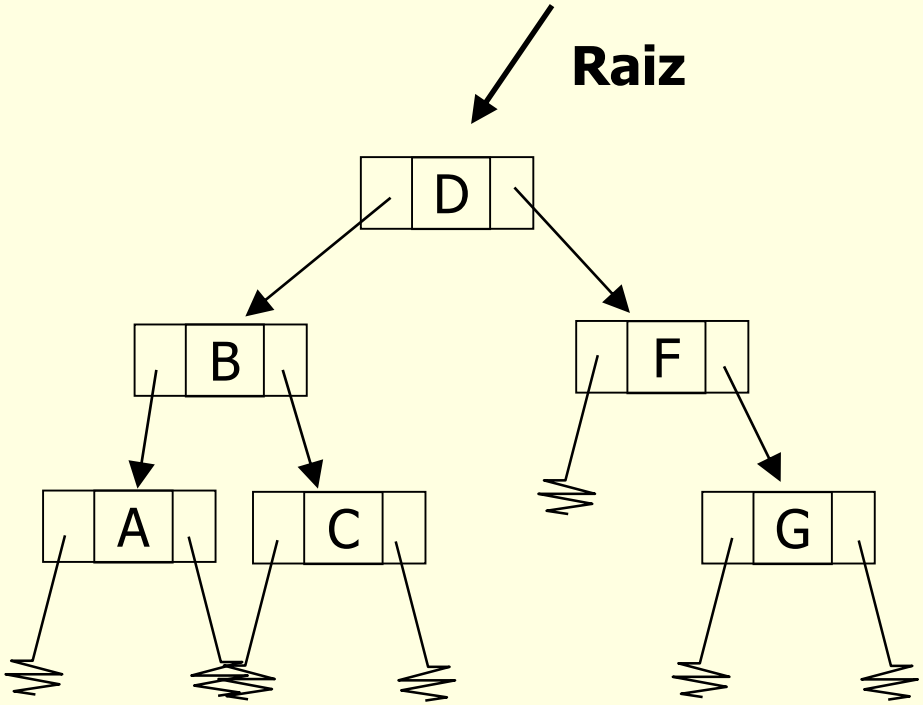
Árvores binárias

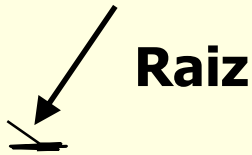
- Declaração

```
typedef char elem;

typedef struct bloco {
        elem info;
        struct bloco *esq, *dir;
} no;

typedef struct {
        no *raiz;
} Arvore;
```

Árvores binárias

- Representação dinâmica e encadeada de uma árvore binária





AB vazia

Árvores binárias

- Implementar o TAD árvore binária: dinâmico e encadeado
  - Já fizemos em aula anterior
    - Criar árvore
    - Verificar se a árvore está vazia
    - Buscar um elemento

Relembrando

```
void cria(Arvore *A) {
     A->raiz=NULL;
}
```

```
int IsEmpty(Arvore *A) {
    if (A->raiz==NULL)
        return 1;
    else return 0;
}
```

```
no* busca(no *p, elem *x) {
    no *aux;
    if (p==NULL)
        return(NULL);
    else if (p->info==*x)
            return(p);
    else {
            aux=busca(p->esq,x);
            if (aux==NULL)
                aux=busca(p->dir,x);
            return(aux);
    }
}
```

Árvores binárias

- Implementar o TAD árvore binária: dinâmico e encadeado
  - Buscar pai de um elemento
  - Inserir elemento à esquerda de outro elemento
  - Inserir elemento à direita de outro elemento
  - Imprimir elementos da árvore
  - Finalizar árvore
  - Determinar altura da árvore

Árvores binárias

- Exercício
  - Simulação da execução da função de cômputo de altura de uma árvore
    - Considere uma árvore simples para simular a execução

Exercício

- Em duplas
  - Esquematize/desenhe/explique como seria a função de remoção de um elemento da árvore
  - Implemente a função de remoção