Aula 12 - Árvores – Conceitos
Árvores Binárias

Prof. Emerson A Marconato

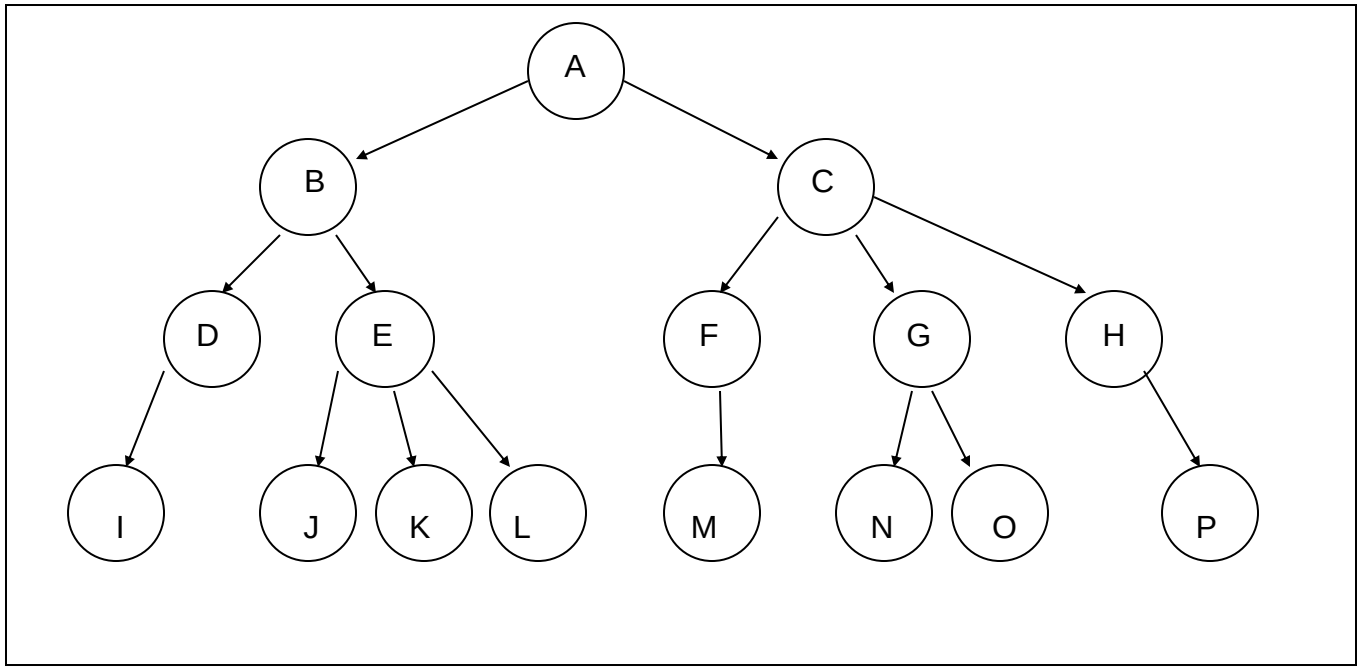
UNIVEM

Árvores - Definição

Uma árvore com tipo base T é:

- ou uma árvore vazia;
- ou um nó do tipo T com um conjunto de árvores distintas associadas ao tipo T, chamadas de sub-árvores. Árvore é uma estrutura homogênea, isto é, todos os nós são do mesmo tipo.

Representação



Prof. Emerson A Marconato
marconato@univem.edu.br

Terminologias

- RAIZ: é o nó que está no topo da árvore. Na árvore acima é o nó que contém a informação ‘A’.

- DESCENDENTES: são os nós que estão ligados abaixo de um nó específico.

- DESCENDENTES DIRETOS: são os nós que estão ligados diretamente ao nó de nível superior.

Terminologias

- ALTURA DA ÁRVORE: é o máximo nível de uma árvore.
- NÍVEL DE UM NÓ: é a distância que o separa da raiz. A raiz tem nível = 0.
- FOLHA: são os nós que não possuem descendentes.
- GRAU DE UM NÓ: é o número de descendentes diretos.
- GRAU DE UMA ÁRVORE: é o máximo grau de seus nós.

Na árvore utilizada como exemplo identifique:

a) quem é a raiz ?

b) quem são as folhas ?

c) quem são os descendentes do nó C ?

d) qual é o grau do nó G ?

e) qual é o grau da árvore ?

f) qual é o nível do nó E ?

g) qual é altura da árvore ?

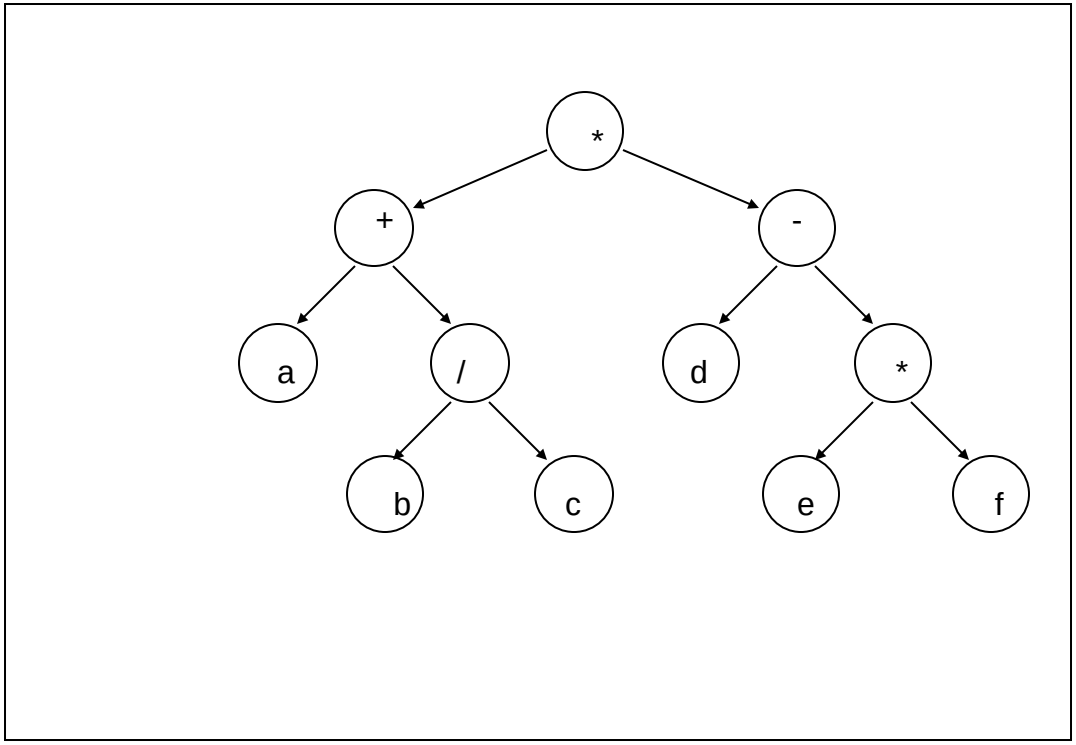
Árvores Binárias

Uma árvore binária ordenada é um conjunto finito de nós que, ou é vazio, ou consiste de uma raiz com 0, 1 ou 2 sub-árvores. Quando existem duas sub-árvores estas são chamadas de sub-árvore à esquerda e sub-árvore à direita. Por exemplo veja a representação da expressão:

(a + b / c) * (d - e * f)

Prof. Emerson A Marconato
marconato@univem.edu.br

Esquematizando...



Prof. Emerson A Marconato
marconato@univem.edu.br

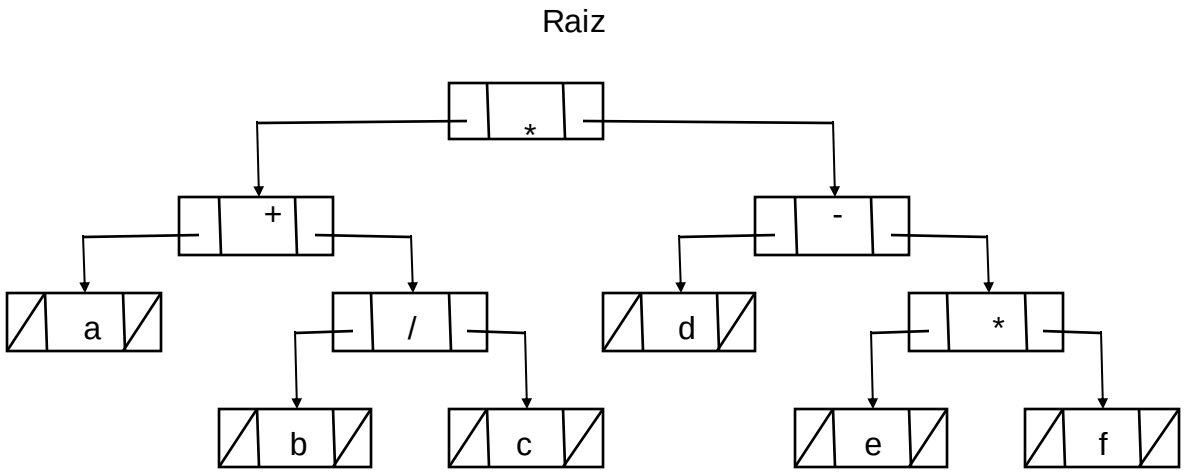
Declaração de um nó em C

Cada nó da árvore binária seria definido em C da seguinte forma:

```
struct no_arvore {
        int info;
        struct no_arvore *esq, *dir;
};

typedef struct no_arvore ARVORE;
```

Esquematizando...



Terminologias...

- Se A é a raiz de uma árvore binária e B é a raiz de sua sub-árvore esquerda ou direita, então diz-se que **A é o pai de B** e que **B é o filho direito ou esquerdo de A**.

- Um nó sem filhos é chamado de **folha**.

- Dois nós são **irmãos** se forem filhos direito e esquerdo do mesmo pai.

Continuando...

- Se todo nó que não é folha numa árvore binária tiver sub-árvores esquerda e direita não vazias, a árvore será considerada **árvore estritamente binária.**

- No exemplo citado, tem-se uma árvore estritamente binária.

Continuando...

- Uma **árvore binária completa** de profundidade (altura) d é a árvore estritamente binária em que todas as folhas estejam no mesmo nível d.

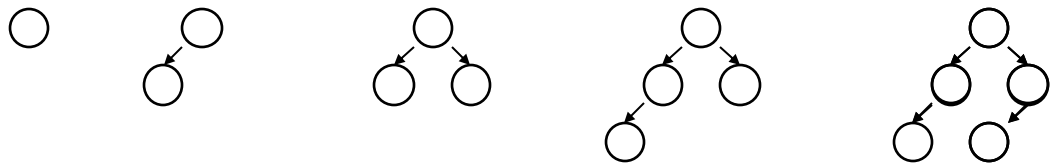
- Uma árvore é dita **perfeitamente balanceada** se para cada nó da árvore, o número de nós em suas sub-árvores à esquerda e à direita diferirem de no máximo 1 nó.

Árvores Perfeitamente Balanceadas

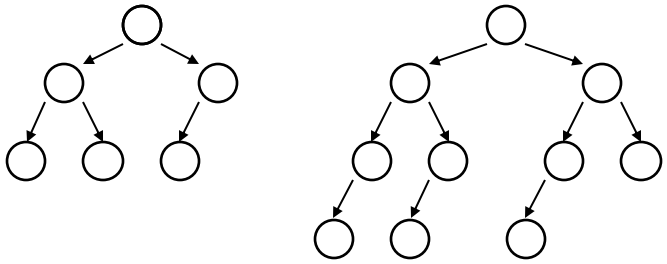
- Construir uma árvore com N elementos é vago, pois não especifica a posição dos nós (podem ser geradas diversas árvores distintas) . Se especificarmos : a construção de uma árvore com altura (profundidade) mínima H, poderíamos construí-la de acordo com o esquema a seguir. Seja N o número de nós da árvore, então:

Esquematizando...

n=1 n=2 n=3 n=4 n=5



n=6 n=10



Total de nós...

- Para estas representações N é o número de nós e existe um nó que é a raiz. A sub-árvore à esquerda possui (N div 2) nós e a sub-árvore à direita possui N - (N div 2) - 1 nós, estas são chamadas de **árvores perfeitamente balanceadas.**

Função para criar uma árvore perfeitamente balanceada com N nós

```
ARVORE *CAPB (int N)
{
   ARVORE *r;
   if (N==0)
      r=NULL;
   else
   {
      r = (ARVORE *) calloc (1, sizeof(ARVORE));
      printf("\nDigite um valor => ");
      scanf("%i",&r->info);
      r->esq = CAPB(N/2);
      r->dir = CAPB(N - N/2 - 1);
   }
   return r;
}
```

Prof. Emerson A Marconato
marconato@univem.edu.br

Função main( )

```
main()
{
   int qtd;
   ARVORE *Raiz;
   printf ("Digite o número de nós para a
   árvore");
   scanf("%i",&qtd);
   Raiz = CAPB(qtd);
    ....
}
```

Função para imprimir a árvore

```
void Imprime(ARVORE *R)
{
   if (R != NULL)
   {
        printf ("%i  ",R->info);
        Imprime (R->esq);
        Imprime (R->dir);
   }
}
```

Exercício

- Elabore uma função que mostre os descendentes diretos de um determinado nó da árvore.