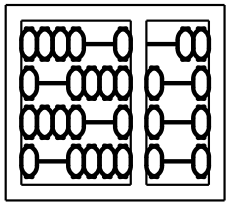


Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC

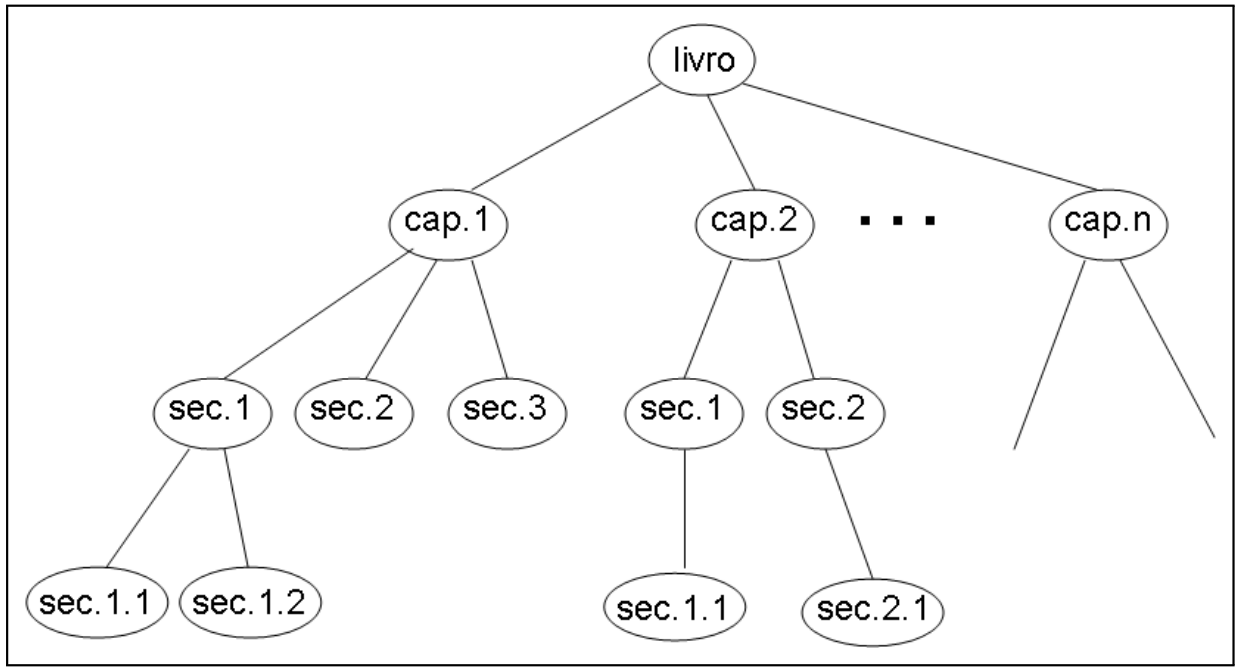
Árvores: Parte 1

Tópicos

1. Introdução
2. Terminologia
3. Árvores binárias
4. Exemplos de funções sobre árvores binárias

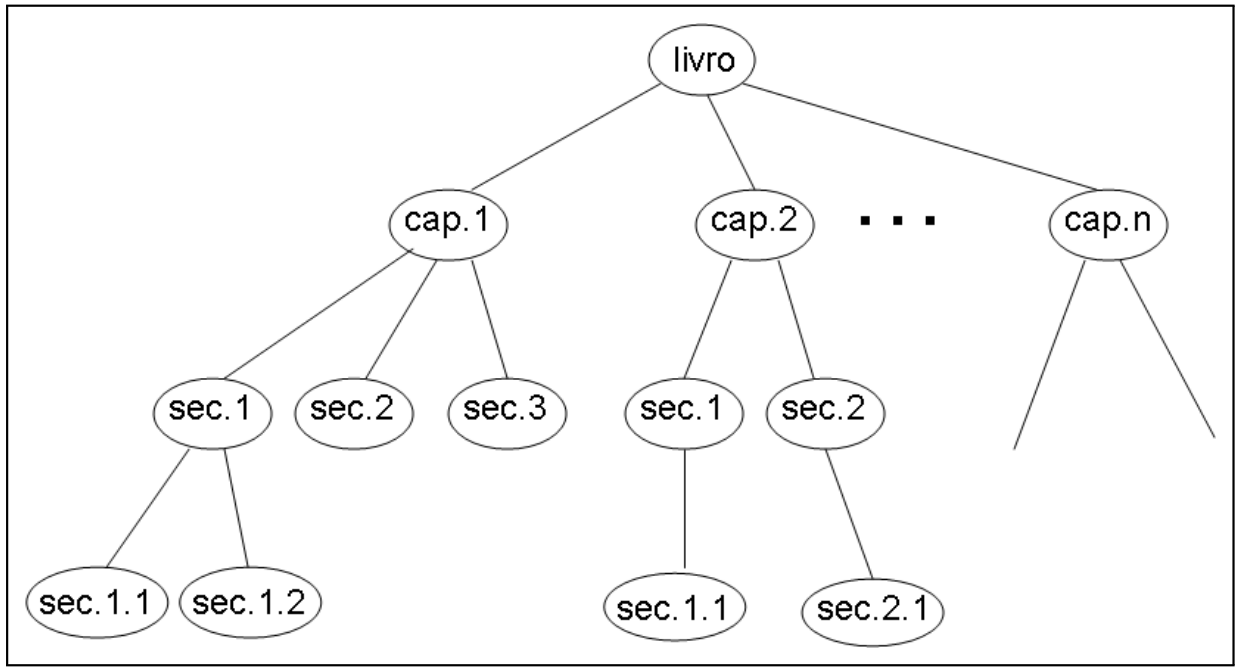
- Representam uma estrutura hierárquica da informação.

– Exemplo:



- Representam uma estrutura hierárquica da informação.

– Exemplo:



Aplicações: Engenharia, Matemática, Administração etc.

- Definição:

– Uma árvore é um conjunto finito de um ou mais nós tal que:

- Definição:

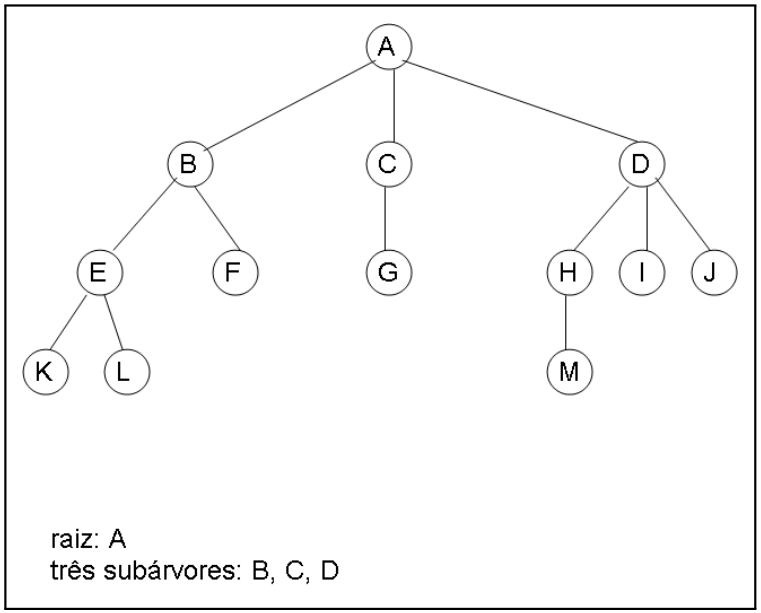
– Uma árvore é um conjunto finito de um ou mais nós tal que:

1. Existe um nó especial chamado *raiz*.
2. Os outros nós estão particionados em $n \geq 0$ conjuntos disjuntos T_1, T_2,..., T_n, em que cada um destes conjuntos é uma árvore (*subárvore* da raiz).

- Definição:

– Uma árvore é um conjunto finito de um ou mais nós tal que:

1. Existe um nó especial chamado *raiz*.
2. Os outros nós estão particionados em $n \geq 0$ conjuntos disjuntos T_1, T_2,..., T_n, em que cada um destes conjuntos é uma árvore (*subárvore* da raiz).



Esta árvore tem 13 nós cujo conteúdo é uma letra. A raiz é *A* e é representada normalmente no topo.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- Terminologia:

– Grau de um nó: é o número de subárvores que ele possui (por exemplo, gr(A) = 3, gr(C) = 1, gr(F) = 0)

– Nós de grau zero são denominados nós terminais ou folhas (K,L,F,G,M,I,J formam o conjunto das folhas da árvore anterior).

– Os outros nós são não terminais.

– As raízes das subárvores de um nó X são os filhos de X. Neste caso, X é o pai (por exemplo, os filhos do nó D são H,I,J. O pai de D é A.

– O grau da árvore é o maior grau dos nós da árvore. No exemplo anterior a árvore tem grau 3.

– Os ancestrais de um nó são todos os nós ao longo do caminho entre a raiz e o nó.

- ... continuação

– O nível ou profundidade de um nó é definido recursivamente da seguinte forma:

1. O nível da raiz é 1.
2. Se um nó está no nível n, então seus filhos estão no nível $n + 1$.

– A profundidade ou altura de uma árvore é definida como sendo o máximo nível de qualquer nó da árvore.

– Uma floresta é um conjunto finito de $n \geq 0$ árvores disjuntas.

- ... continuação

– O nível ou profundidade de um nó é definido recursivamente da seguinte forma:

1. O nível da raiz é 1.
2. Se um nó está no nível n, então seus filhos estão no nível $n + 1$.

– A profundidade ou altura de uma árvore é definida como sendo o máximo nível de qualquer nó da árvore.

– Uma floresta é um conjunto finito de $n \geq 0$ árvores disjuntas.

- ... continuação

– O nível ou profundidade de um nó é definido recursivamente da seguinte forma:

1. O nível da raiz é 1.

2. Se um nó está no nível n, então seus filhos estão no nível $n + 1$.

– A profundidade ou altura de uma árvore é definida como sendo o máximo nível de qualquer nó da árvore.

– Uma floresta é um conjunto finito de $n \geq 0$ árvores disjuntas.

- ... continuação

– O nível ou profundidade de um nó é definido recursivamente da seguinte forma:

1. O nível da raiz é 1.

2. Se um nó está no nível n, então seus filhos estão no nível $n + 1$.

– A profundidade ou altura de uma árvore é definida como sendo o máximo nível de qualquer nó da árvore.

– Uma floresta é um conjunto finito de $n \geq 0$ árvores disjuntas.

- ... continuação

– O nível ou profundidade de um nó é definido recursivamente da seguinte forma:

1. O nível da raiz é 1.

2. Se um nó está no nível n, então seus filhos estão no nível $n + 1$.

– A profundidade ou altura de uma árvore é definida como sendo o máximo nível de qualquer nó da árvore.

– Uma floresta é um conjunto finito de $n \geq 0$ árvores disjuntas.

Árvores binárias

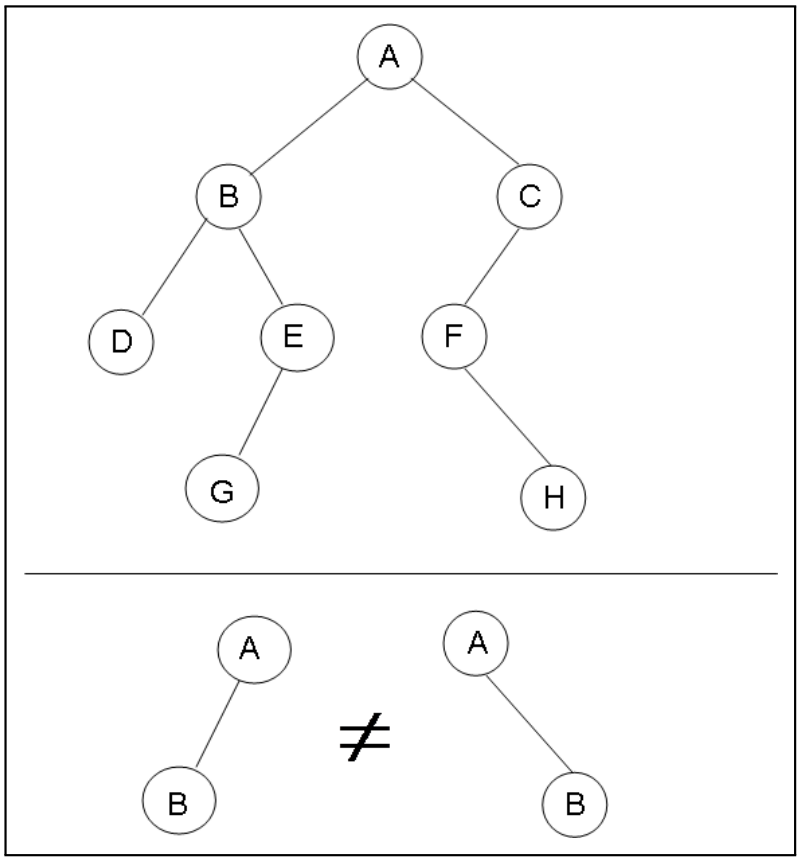
- Definição: conjunto de nós que pode ser vazio ou consiste de:

  1. Um nó denominado raiz.
  2. Duas árvores binárias disjuntas denominadas subárvores à *esquerda* e à *direita* da raiz.

Árvores binárias

- Definição: conjunto de nós que pode ser vazio ou consiste de:

  1. Um nó denominado raiz.
  2. Duas árvores binárias disjuntas denominadas subárvores à *esquerda* e à *direita* da raiz.



- Outras propriedades:

1. Lema: O número máximo de nós no nível i de uma árvore binária é 2^{i-1}, $i \geq 1$.

2. Corolário: Uma árvore de profundidade k tem no máximo $2^k - 1$ nós.

- Outras propriedades:

1. Lema: O número máximo de nós no nível i de uma árvore binária é 2^{i-1}, $i \geq 1$.

2. Corolário: Uma árvore de profundidade k tem no máximo $2^k - 1$ nós.

- Árvore binária completa:

– Uma árvore binária completa de nível k é aquela em que:

- Árvore binária completa:

– Uma árvore binária completa de nível k é aquela em que:

1. Cada nó de nível k é uma folha.

- Árvore binária completa:

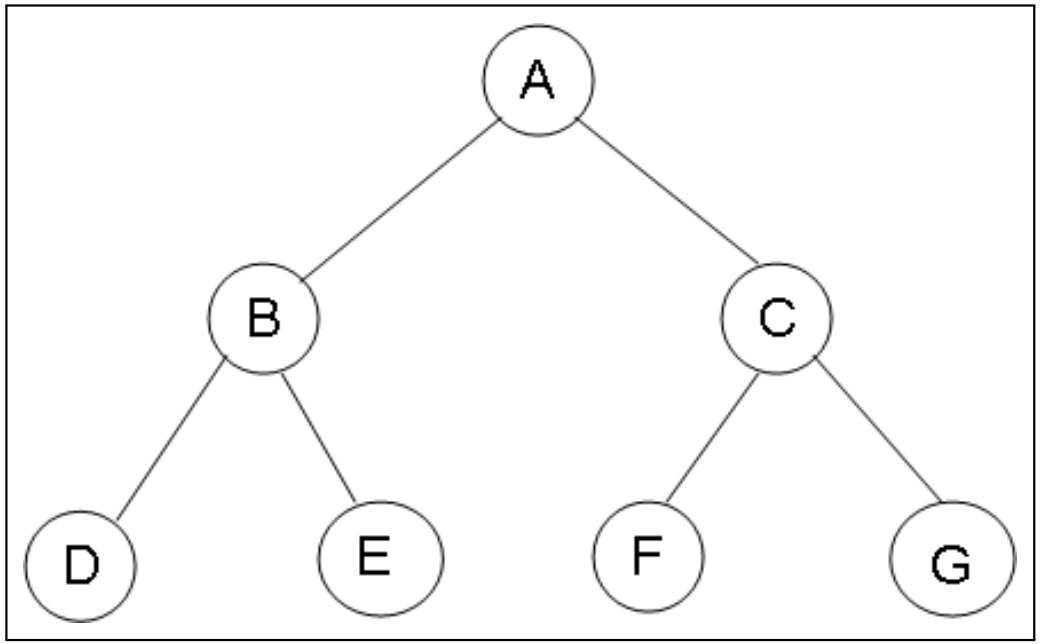
– Uma árvore binária completa de nível k é aquela em que:

1. Cada nó de nível k é uma folha.
2. Cada nó de nível menor do que k tem subárvores não vazias à esquerda e à direita.

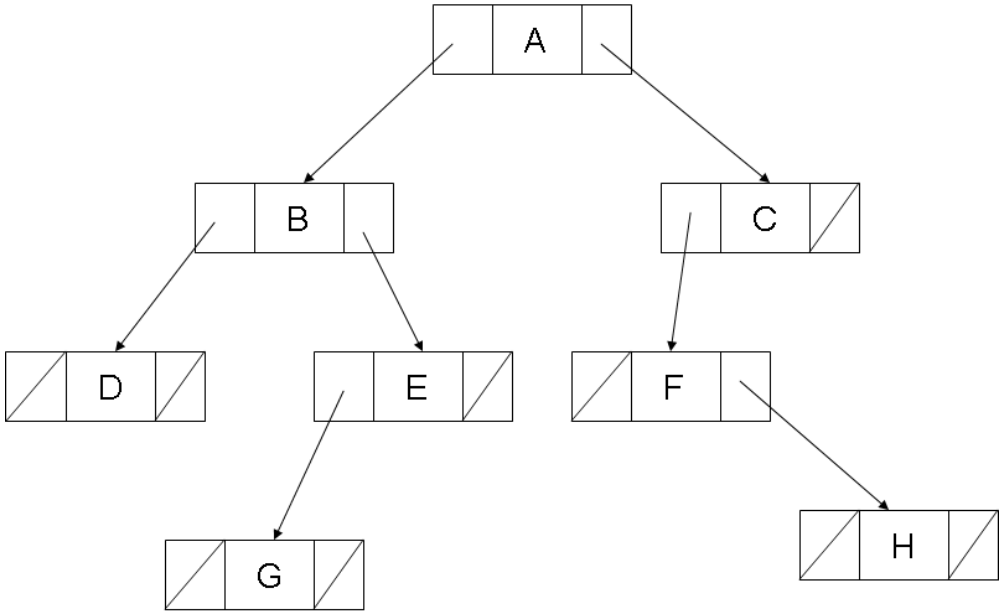
- Árvore binária completa:

– Uma árvore binária completa de nível k é aquela em que:

1. Cada nó de nível k é uma folha.
2. Cada nó de nível menor do que k tem subárvores não vazias à esquerda e à direita.



- Implementação de árvores binárias



- Estrutura:

```
typedef struct No {
  int info ;  /* exemplo do conteúdo de cada no */
  struct No *esq, *dir ;  /* apontadores para os filho */
}  ArvBin, NoArvBin ;
```

– O **endereço** de uma árvore binária é o endereço de sua raiz (o endereço de uma árvore vazia é *NULL*).

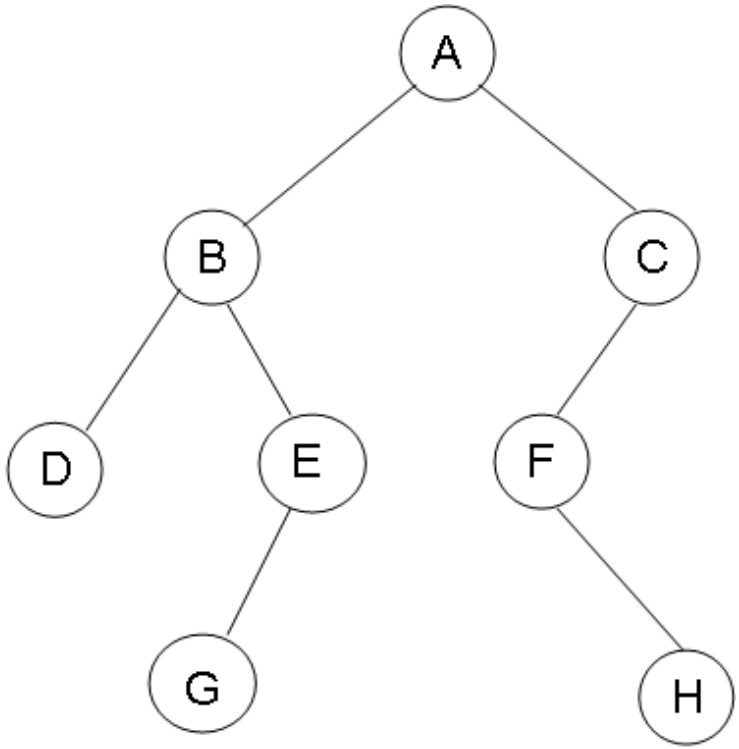
- Estrutura:

```
typedef struct No {
  int info ;  /* exemplo do conteúdo de cada no */
  struct No *esq, *dir ;  /* apontadores para os filho */
}  ArvBin, NoArvBin ;
```

– O **endereço** de uma árvore binária é o endereço de sua raiz (o endereço de uma árvore vazia é *NULL*).

- Propriedade:

– Uma árvore binária com n nós utiliza $n - 1$ apontadores não nulos.



Nós: 8
Apontadores não nulos: 7

Exemplos de funções

- Criar uma árvore vazia:

```
ArvBin *criaVazia (void) {
  return NULL ;
}
```

Exemplos de funções

- Criar um nó apontado por *p*:

```
void criaNo (ArvBin **p, int x) {
  *p = (NoArvBin *)malloc(sizeof(NoArvBin)) ;
  (*p)->info = x ;
  (*p)->esq = NULL ;
  (*p)->dir = NULL ;
```

Exemplos de funções

- Adicionar um nó à esquerda de um nó p:

```
void AdicionaEsq (ArvBin *p, int x) {
 NoArvBin *q ;
 if (p == NULL) ''Erro'' ;
 else {
   CriaNo (&q, x) ;
   p->esq = q ;
 }
}
```

Exemplos de funções

- Inserir um nó à esquerda de um nó p:

```
void InsereEsq (ArvBin *p, int x) {
  NoArvBin *q ;
  q = (NoArvBin *)malloc(sizeof(NoArvBin)) ;
  q->info = x ;
  q->esq = p->esq ;
  q->dir = NULL ;
  p->esq = q ;
}
```

Exemplos de funções

• A operação de se construir uma árvore não vazia pode ser representada, ainda, pela seguinte função:

```
ArvBin *criaArv (int x, ArvBin *NoEsq, ArvBin *NoDir) {
  ArvBin *p = (ArvBin *)malloc(sizeof(ArvBin)) ;
  p->info = x ;
  p->esq = NoEsq ;
  p->dir = NoDir ;

  return p ;    /* retorna nó raiz criado */
}
```

– Esta função e a função *criaVazia* anterior representam os dois casos de definição recursiva de árvores binárias.

Exemplos de funções

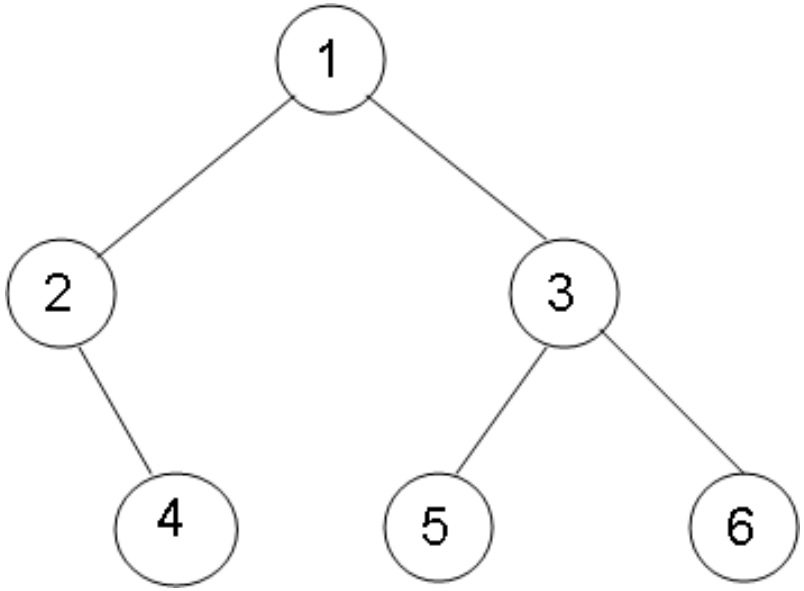
• A operação de se construir uma árvore não vazia pode ser representada, ainda, pela seguinte função:

```
ArvBin *criaArv (int x, ArvBin *NoEsq, ArvBin *NoDir) {
  ArvBin *p = (ArvBin *)malloc(sizeof(ArvBin)) ;
  p->info = x ;
  p->esq = NoEsq ;
  p->dir = NoDir ;

  return p ;    /* retorna nó raiz criado */
}
```

– Esta função e a função *criaVazia* anterior representam os dois casos de definição recursiva de árvores binárias.

• A partir das duas funções mencionadas anteriormente, a árvore abaixo pode ser construída da seguinte forma:



...

```
/* subarvore 4 */
ArvBin *a4 = criaArv(4, criaVazia(), criaVazia()) ;
/* subarvore 2 */
ArvBin *a2 = criaArv(2, criaVazia(), a4) ;
/* subarvore 5 */
ArvBin *a5 = criaArv(5, criaVazia(), criaVazia()) ;
/* subarvore 6 */
ArvBin *a6 = criaArv(6, criaVazia(), criaVazia()) ;
/* subarvore 3 */
ArvBin *a3 = criaArv(3, a5, a6) ;
/* subarvore 1 */
ArvBin *a1 = criaArv(1, a2, a3) ;
```

- ... ou com uma única atribuição:

```
ArvBin *a1 = criaArv(1,                    /*lado esq. de a1 */
                criaArv(2,
                criaVazia(),
                criaArv(4, criaVazia(), criaVazia()))
                ,
                criaArv(3,                /* lado dir. de a1 */
                criaArv(5, criaVazia(), criaVazia()),
                criaArv(6, criaVazia(), criaVazia()))
                ) ;
```

Exemplos de funções

- Imprimir o conteúdo de uma árvore:

```
void arv_imprime( ArvBin *p) {
  if(p != NULL) {
    printf(''%d '', p->info) ;     /* imprime raiz */
    arv_imprime(p->esq) ;          /* imprime no esquerdo */
    arv_imprime(p->dir) ;          /* imprime no direito */
  }
}
```