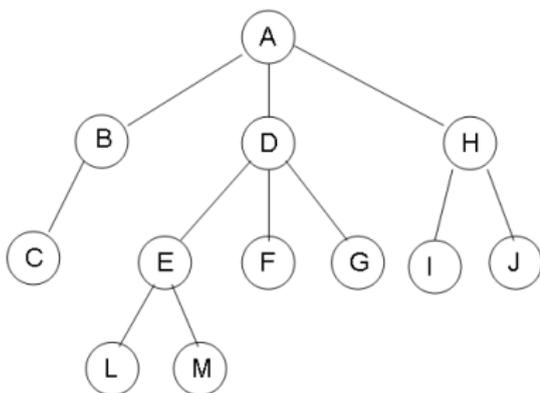


Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC

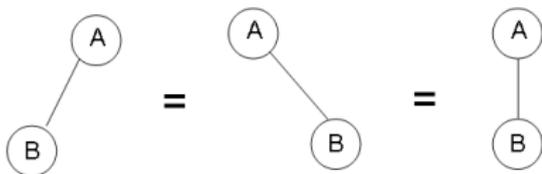
Árvores Gerais

- 1 Árvores Gerais
- 2 Florestas
- 3 Árvores Digitais
- 4 Árvores Binárias Digitais

- Definição: Uma *árvore geral* T é um conjunto finito e não vazio de objetos (nós ou vértices), tal que T consiste de um nó distinto, denominado raiz de T , e de $m \geq 0$ árvores disjuntas, T_1, T_2, \dots, T_m , chamadas subárvores de T .

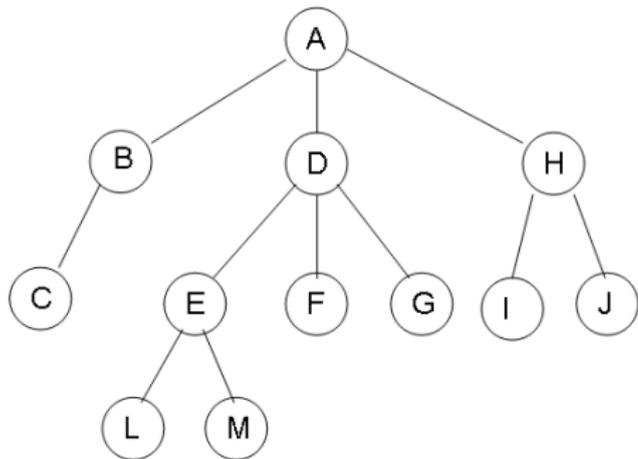


- Observe que não existe o conceito de árvore vazia, como no caso binário e, portanto:

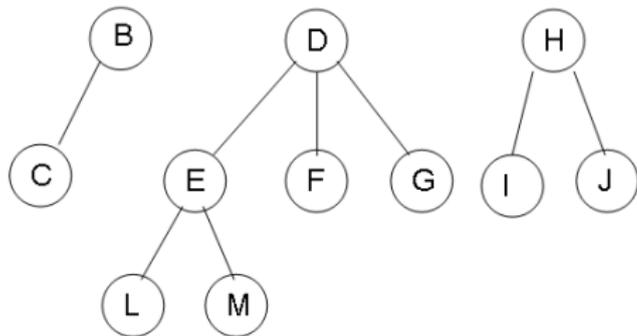


- Definição: Uma floresta F é uma sequência finita de árvores ($\overline{F} = T_1, T_2, \dots, T_m$), $m \geq 0$ e cada T_i representa uma árvore da floresta.
 - Observe que esta definição é recursiva pois uma árvore pode ser vista como a sua raiz juntamente com a floresta das suas subárvores.

- Exemplo: Retirando-se a raiz A da árvore anterior obtém-se uma floresta com raízes B , D , e H .



- Exemplo: Retirando-se a raiz A da árvore anterior obtém-se uma floresta com raízes B , D , e H .



- Representação de árvores gerais:

– Exemplo 1: Considera vetores de apontadores, prevendo-se um número máximo *GrauMax* de filhos em cada nó.

```
#define GRAU_MAX 10
typedef struct NoArvGeral *ArvGeral ;
typedef struct NoArvGeral {
    int info ;          /* o campo info do nó */
    int grau ;         /* o grau do nó */
    ArvGeral filhos[GRAU_MAX] ;    /* os filhos do nó */
} NoArvGeral ;
...
p = malloc(sizeof(NoArvGeral)) ;
```

⇒ Eventual desperdício de memória.

- Representação de árvores gerais:

– Exemplo 1: Considera vetores de apontadores, prevendo-se um número máximo *GrauMax* de filhos em cada nó.

```
#define GRAU_MAX 10
typedef struct NoArvGeral *ArvGeral ;
typedef struct NoArvGeral {
    int info ;          /* o campo info do nó */
    int grau ;         /* o grau do nó */
    ArvGeral filhos[GRAU_MAX] ;    /* os filhos do nó */
} NoArvGeral ;
...
p = malloc(sizeof(NoArvGeral)) ;
```

⇒ **Eventual desperdício de memória.**

– Exemplo 2: O número de componentes do vetor filho é indicado na alocação dinâmica.

```
typedef struct NoArvGeral  *ArvGeral ;
typedef struct NoArvGeral {
    int info ;           /* o campo info do nó */
    int grau ;          /* o grau do nó */
    ArvGeral filhos[1] ;
} NoArvGeral ;

...
p = malloc(sizeof(NoArvGeral)+(grau-1)*sizeof(ArvGeral)) ;
```

⇒ Maior dificuldade de manipulação em caso de operações que modifiquem o grau de um nó, tais como inserção e remoção.

- Exemplo 2: O número de componentes do vetor filho é indicado na alocação dinâmica.

```
typedef struct NoArvGeral  *ArvGeral ;
typedef struct NoArvGeral {
    int info ;           /* o campo info do nó */
    int grau ;          /* o grau do nó */
    ArvGeral filhos[1] ;
} NoArvGeral ;

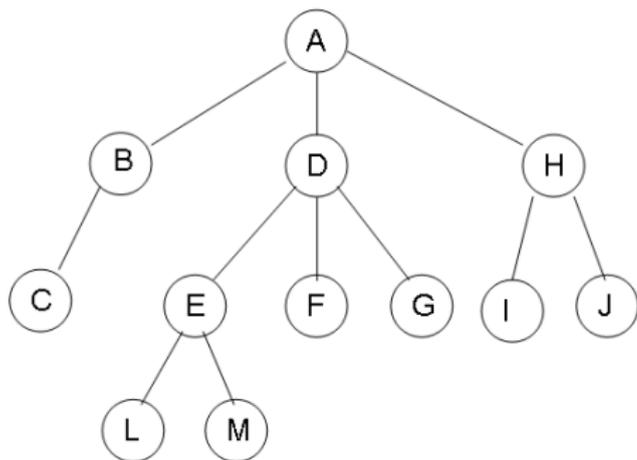
...
p = malloc(sizeof(NoArvGeral)+(grau-1)*sizeof(ArvGeral)) ;
```

⇒ **Maior dificuldade de manipulação em caso de operações que modifiquem o grau de um nó, tais como inserção e remoção.**

- Alternativa: Representação binária da árvore geral:

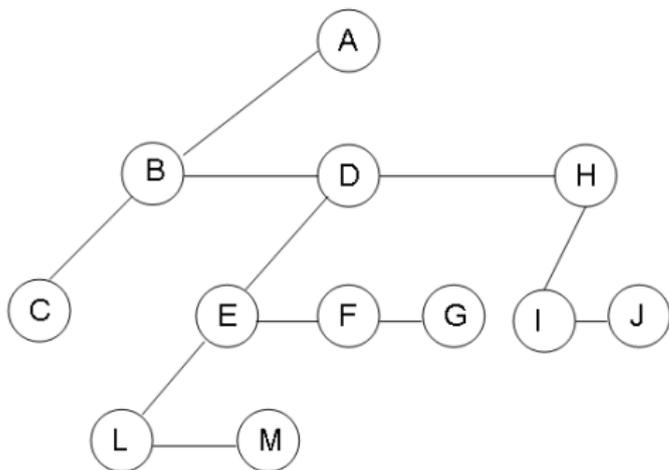
- Alternativa: Representação binária da árvore geral:

Passo 1: O filho mais à esquerda permanece ligado à esquerda do seu respectivo pai. Seus irmãos são colocados no mesmo nível.

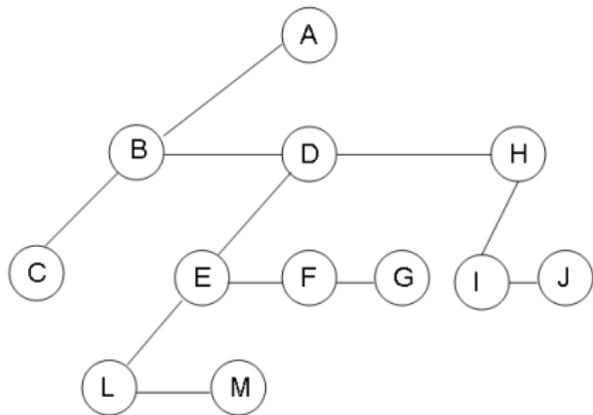


- Alternativa: Representação binária da árvore geral:

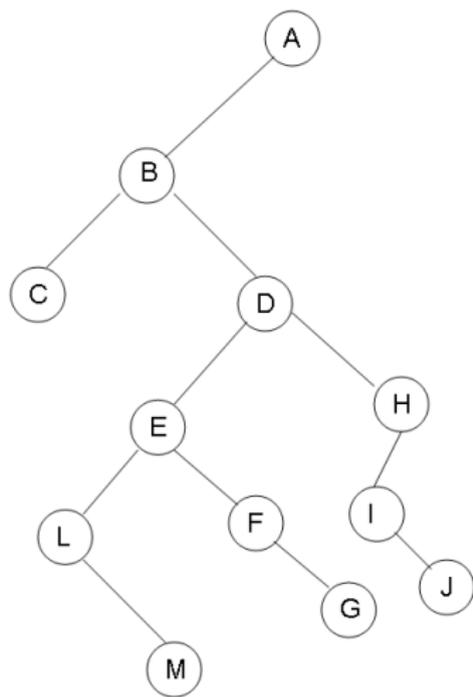
Passo 1: O filho mais à esquerda permanece ligado à esquerda do seu respectivo pai. Seus irmãos são colocados no mesmo nível.



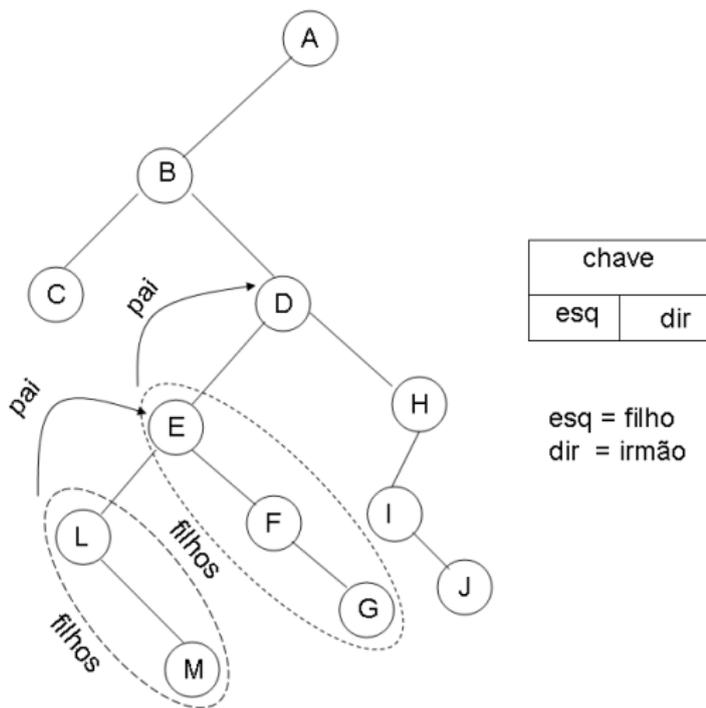
Passo 2: Obtém-se a árvore binária girando-se as **linhas horizontais** da representação anterior de -45° .



Passo 2: Obtém-se a árvore binária girando-se as **linhas horizontais** da representação anterior de -45° .



Passo 2: Obtém-se a árvore binária girando-se as **linhas horizontais** da representação anterior de -45° .



- Representação recursiva de florestas:

– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

- Representação recursiva de florestas:

– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

- Representação recursiva de florestas:

– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

- Representação recursiva de florestas:

– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

- Representação recursiva de florestas:

– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

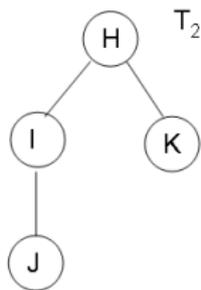
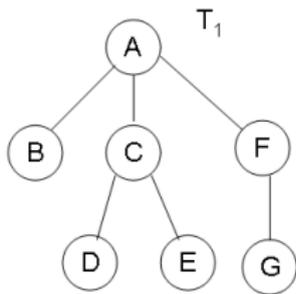
1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

- Representação recursiva de florestas:

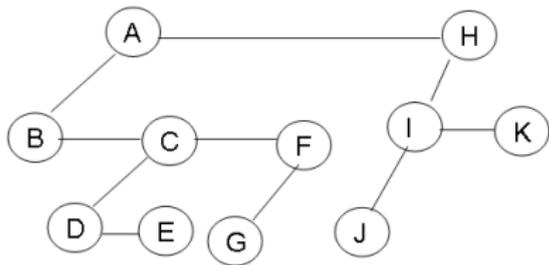
– Seja $F = (T_1, T_2, \dots, T_m)$ uma floresta com $m \geq 0$. A árvore binária $B(F)$, representando F , é dada por:

1. Árvore vazia se F é uma árvore vazia ($m = 0$).
2. Árvore binária cuja raiz contém a mesma informação da raiz de T_1 ;
 1. cuja subárvore esquerda é dada por $B((T_{11}, T_{12}, \dots, T_{1m1}))$, em que $(T_{11}, T_{12}, \dots, T_{1m1})$ é a floresta das subárvores de T_1 ;
 2. cuja subárvore direita é dada por $B((T_2, \dots, T_m))$.

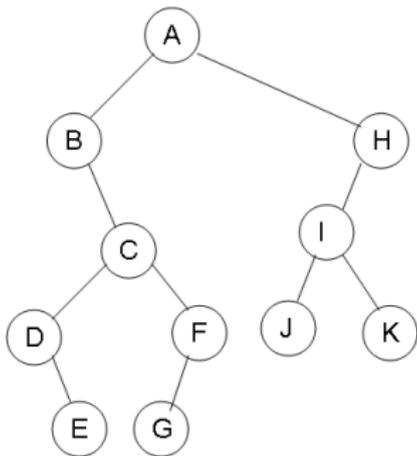
- Exemplo:



- Exemplo:



- Exemplo:



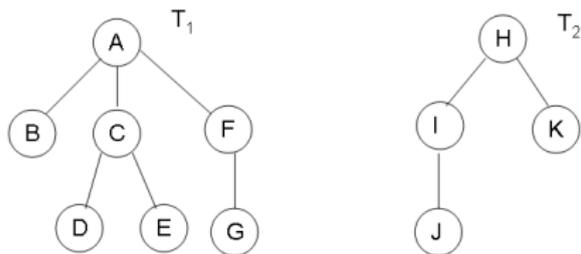
- Percursos de floresta:

- Percursos de floresta:
 - Pré-ordem de floresta:
 - 1 visitar a raiz da 1ª árvore T_1 da floresta.
 - 2 percorrer em pré-ordem a floresta das subárvores de T_1 .
 - 3 percorrer em pré-ordem as florestas T_2, \dots, T_n .

- Percursos de floresta:

- Pré-ordem de floresta:

- 1 visitar a raiz da 1ª árvore T_1 da floresta.
- 2 percorrer em pré-ordem a floresta das subárvores de T_1 .
- 3 percorrer em pré-ordem as florestas T_2, \dots, T_n .



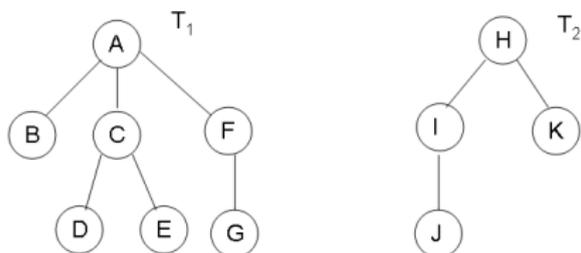
pré-ordem: A B C D E F G H I J K

- Percursos de floresta:
- Pós-ordem de floresta:
 - 1 percorrer em pós-ordem a floresta das subárvores da 1ª árvore T_1 .
 - 2 percorrer em pós-ordem as florestas T_2, \dots, T_n .
 - 3 visitar a raiz de T_1 .

- Percursos de floresta:

- Pós-ordem de floresta:

- 1 percorrer em pós-ordem a floresta das subárvores da 1ª árvore T_1 .
- 2 percorrer em pós-ordem as florestas T_2, \dots, T_n .
- 3 visitar a raiz de T_1 .



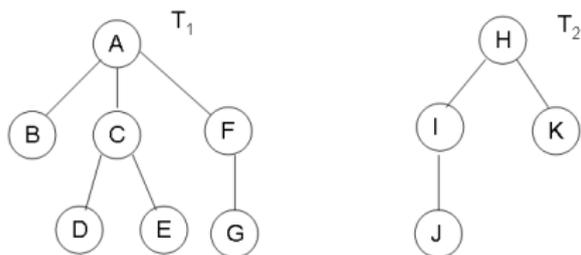
pós-ordem: E D G F C B J K I H A

- Percursos de floresta:
- Inordem de floresta:
 - ① percorrer em inordem a floresta das subárvores da 1ª árvore T_1 .
 - ② visitar a raiz de T_1 .
 - ③ percorrer em inordem as florestas T_2, \dots, T_n .

- Percursos de floresta:

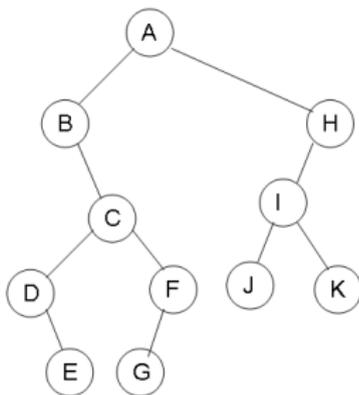
- Inordem de floresta:

- 1 percorrer em inordem a floresta das subárvores da 1ª árvore T_1 .
- 2 visitar a raiz de T_1 .
- 3 percorrer em inordem as florestas T_2, \dots, T_n .



inordem: B D E C G F A J I K H

- Todos os percursos anteriores correspondem àqueles da representação binária.



pré-ordem: A B C D E F G H I J K

pós-ordem: E D G F C B J K I H A

inordem: B D E C G F A J I K H