

CES-11

Algoritmos e Estruturas de Dados

Carlos Alberto Alonso Sanches
Juliana de Melo Bezerra

CES-11

- Árvores
 - Conceito de árvore
 - Definição recursiva de árvore
 - Definições
 - Representações de árvores
 - Ordenação dos nós de uma árvore

CES-11

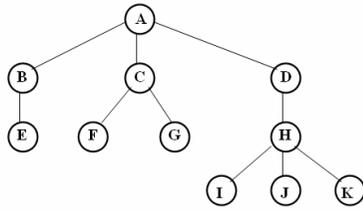
- Árvores
 - **Conceito de árvore**
 - Definição recursiva de árvore
 - Definições
 - Representações de árvores
 - Ordenação dos nós de uma árvore

Conceito de árvore

- Tanto as pilhas como as filas são estruturas lineares, isto é, de uma única dimensão.
- Na sua implementação, as listas ligadas possibilitam maior flexibilidade que os vetores, mas mesmo assim não permitem a representação hierárquica de dados.
- Árvores (*trees*) são estruturas hierárquicas, formadas por vértices e arestas. Ao contrário das árvores naturais, são representadas de cima para baixo: a raiz está no topo e as folhas na base.

Conceito de árvore

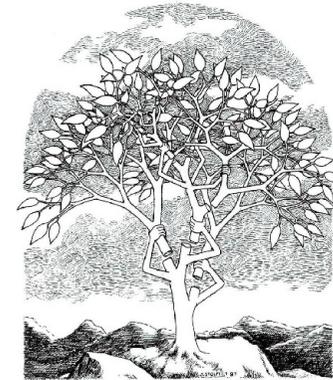
- Cada elemento:
 - Tem um único pai (exceto a raiz)
 - Pode ter vários filhos
 - Não pode ser pai de nenhum ancestral



- A raiz é o único nó que não possui ancestrais
 - Nó A
- As folhas são os nós sem filhos
 - Nós E, F, G, I, J, K

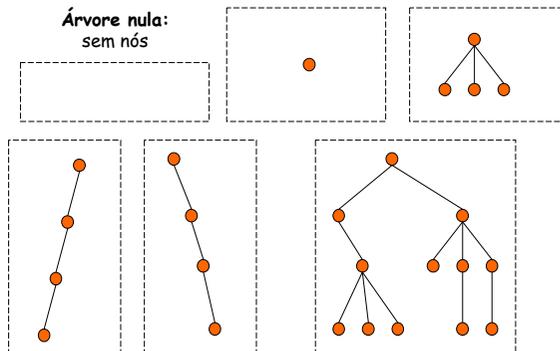
Conceito de árvore

- Um nó não pode ter um ancestral como filho
 - Caso contrário, surgiria um ciclo...
- Desse modo, a árvore é uma estrutura recursiva: cada filho é também raiz de uma outra árvore



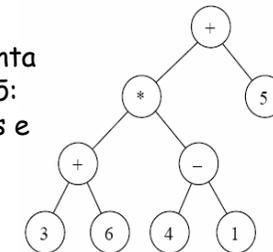
Conceito de árvore

- Exemplos:



Conceito de árvore

- O número de filhos por nó e as informações armazenadas diferenciam os tipos de árvores.
- A árvore ao lado representa a expressão $(3+6)*(4-1)+5$: as folhas possuem valores e os nós intermediários, operadores matemáticos.

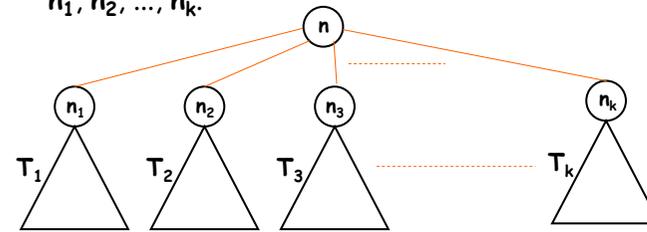


CES-11

- Árvores
 - Conceito de árvore
 - Definição recursiva de árvore
 - Definições
 - Representações de árvores
 - Ordenação dos nós de uma árvore

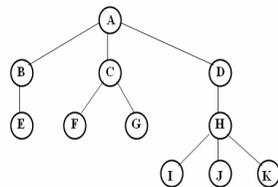
Definição recursiva de árvore

- Um único nó, por si mesmo, é também uma árvore sem filhos, onde ele é a raiz.
- Seja n um nó e T_1, T_2, \dots, T_k árvores de raízes n_1, n_2, \dots, n_k , respectivamente. Pode-se construir uma nova árvore tornando n pai de n_1, n_2, \dots, n_k .



Definição recursiva de árvore

- Comprovação de que a figura abaixo é uma árvore:
 - I, J, K são árvores sem filhos
 - H é raiz de uma nova árvore
 - F e G são árvores sem filhos
 - C é raiz de uma nova árvore
 - E é uma árvore sem filhos
 - B é raiz de uma nova árvore
 - Como B, C e D são árvores, A é raiz de uma nova árvore



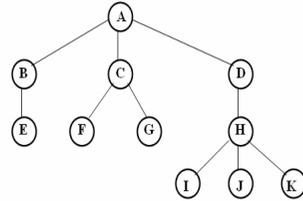
CES-11

- Árvores
 - Conceito de árvore
 - Definição recursiva de árvore
 - Definições
 - Representações de árvores
 - Ordenação dos nós de uma árvore

Definições

- Filho esquerdo de um nó é o seu único filho ou o primeiro filho mais à esquerda.

- B é o filho esquerdo de A
- F é o filho esquerdo de C
- H é o filho esquerdo de D

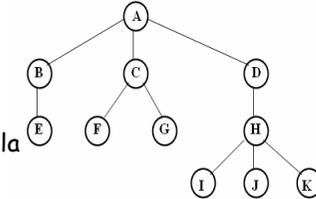


Definições

- Irmão direito de um nó é o irmão imediatamente à direita desse nó.

- Nó sem irmão direito é chamado de caçula.

- C é o irmão direito de B
- K é o irmão direito de J
- H, G, K são caçulas
- E é filho esquerdo e caçula



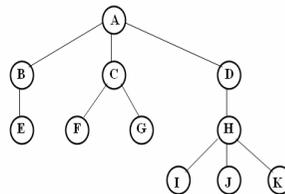
Definições

- Caminho de um nó n_1 a outro n_k : sequência de nós $n_1, n_2, n_3, \dots, n_{k-1}, n_k$ tais que n_i é pai de n_{i+1} , com $1 \leq i < k$.

- Ex: A, AB, ABE, AC, B, ACG, CF, ADHJ, ADHK, DH, DHI, etc.

- O comprimento de um caminho é a sua quantidade de arestas.

- Ex: $|A| = 0$, $|AB| = 1$, $|DHI| = 2$
- É o número de nós menos 1



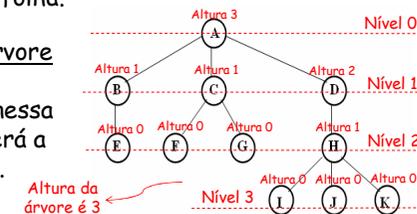
Definições

- Cada nó pode ser alcançado a partir da raiz através de um único caminho.

- O nível (ou profundidade) de um nó é o comprimento do caminho da raiz até esse nó.

- Altura de um nó é o comprimento do mais longo caminho desse nó a alguma folha.

- A altura de uma árvore não vazia é o nível máximo de um nó nessa árvore (ou seja, será a altura da sua raiz).

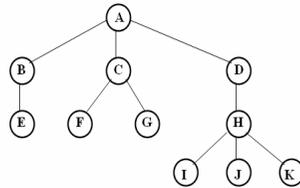


Definições

- Se há um caminho de n_i a n_j , então n_i é ancestral de n_j e n_j é descendente de n_i .
- Um nó é ancestral e descendente de si mesmo.
- Ancestral próprio ou descendente próprio de um nó é um ancestral ou descendente, respectivamente, distinto desse mesmo nó.

- Na árvore ao lado:

- Ancestrais próprios de J: A, D, H
- Descendentes próprios de D: H, I, J, K



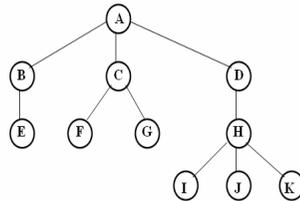
Definições

- Podemos redefinir raiz e folha com os conceitos de ancestral e descendente próprios:
 - Raiz: nó que não possui ancestral próprio
 - Folha: nó que não possui descendente próprio (também chamado nó terminal)
- O grau de um nó é o número de seus filhos.
 - É o número de sub-árvores disjuntas desse nó
 - As folhas têm grau nulo
- O grau de uma árvore é o máximo entre os graus de seus nós.

Definições

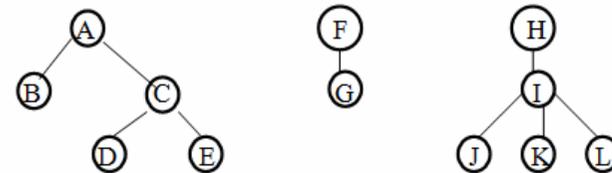
- Um nó que não é folha é chamado de nó interno ou nó não-terminal.
- Irmãos: filhos de um mesmo nó.
- Exemplos na árvore ao lado:

- Grau de C? 2
- Grau de G? 0
- Grau da árvore? 3
- Quais os irmãos de B? C e D
- Quais os irmãos de E? Não há
- Quais os nós não-terminais? A, B, C, D e H



Definições

- Um conjunto de árvores é chamado de floresta.
- Exemplo:



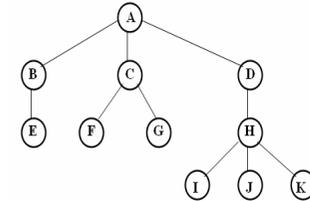
CES-11

Árvores

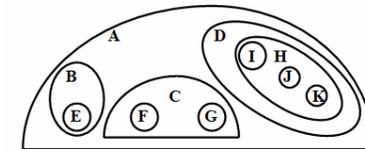
- Conceito de árvore
- Definição recursiva de árvore
- Definições
- Representações de árvores
- Ordenação dos nós de uma árvore

Representações de árvores

a) Forma convencional



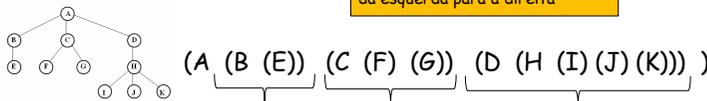
b) Diagrama de conjuntos



Representações de árvores

c) Forma parentética

Coloca-se entre parêntesis a raiz, seguida das formas parentéticas de suas sub-árvores, ordenadas da esquerda para a direita

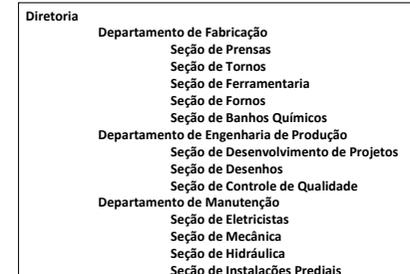


Identificação recursiva de uma forma parentética:

- Sendo c um caractere genérico, (c) é uma forma parentética correta.
- Se $a_1, a_2, a_3, \dots, a_n$ são formas parentéticas corretas, com $n > 0$, então $(c a_1 a_2 a_3 \dots a_n)$ também será.

Representações de árvores

d) Forma tabulada



- Cada nó aparece numa linha, e seus filhos são listados com uma tabulação a mais que a desse nó.

- Ex: organograma de uma empresa, índice de livros, etc.

Representações de árvores

e) Forma numerada
(ou *itemizada*)

```
1 - Estruturas de dados
1.1 - Listas lineares
1.1.1 - Estrutura contígua
1.1.2 - Estrutura encadeada
1.1.3 - Pilhas e filas
1.2 - Árvores
1.2.1 - Definições
1.2.2 - Estruturas de dados
1.2.3 - Árvores binárias
1.3 - Grafos
1.3.1 - Grafos orientados
1.3.2 - Grafos não orientados
```

- Semelhante à anterior: a numeração de um nó tem como prefixo o número de seu pai, e como sufixo um número que o diferencie dos irmãos.

CES-11

- **Árvores**
 - Conceito de árvore
 - Definição recursiva de árvore
 - Definições
 - Representações de árvores
 - **Ordenação dos nós de uma árvore**

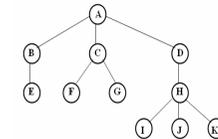
Ordenação dos nós de uma árvore

- a) Ordenação dos filhos de um nó
 - Os filhos de um nó são ordenados da esquerda para a direita
 - Por exemplo, as duas árvores abaixo têm o mesmo pai e os mesmos filhos, mas são diferentes:



Ordenação dos nós de uma árvore

- b) Extensão da ordenação da esquerda para a direita
 - Se X e Y são irmãos e X está à direita de Y, então todos os descendentes de X estão à direita de Y e dos seus descendentes.
 - O caminho da raiz até um determinado nó separa os nós que estão à sua esquerda dos que estão à sua direita.
 - Exemplo: C e D são irmãos e C está à esquerda de D
 - F e G estão à esquerda de D, H, I, J e K
 - J não está nem à direita nem à esquerda de H, D, e A



Ordenação dos nós de uma árvore

c) Ordenação de todos os nós de uma árvore

- Existem formas de se ordenar e de se percorrer sistematicamente todos os nós de uma árvore:

- Ordenação ou percurso por nível
- Ordenação ou percurso em pré-ordem
- Ordenação ou percurso em pós-ordem
- Ordenação ou percurso em ordem central (ou in-ordem)

Busca em largura

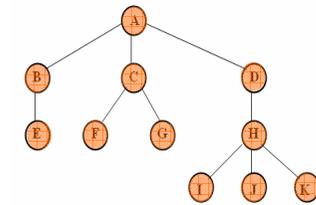
Busca em profundidade

Percurso por nível (ou largura)

Passos:

- Primeiramente, visita-se a raiz
- Depois visitam-se todos os filhos da raiz, da esquerda para a direita
- Depois os netos, depois os bisnetos da raiz, e assim por diante...

- Também é chamado percurso ou busca em largura



A B C D E F G H I J K

Percurso por nível (ou largura)

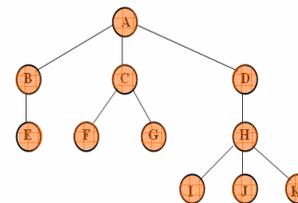
Implementação com uso de fila:

- Depois que um nó é visitado, seus filhos (da esquerda para a direita) são colocados no final da fila.
- O próximo nó a ser visitado é o que está no início da fila.
- Termina quando a fila fica vazia.

```

void OrdenacaoPorNivel(raiz)
{
  Fila q;
  node p = raiz;
  if (p != null) {
    enqueue(q,p);
    while (!isEmpty(q)){
      p = first(q);
      dequeue(q);
      escreve(p);
      Para todo f filho de p
        enqueue(q, f);
    }
  }
}
  
```

Percurso por nível (ou largura)



Fila: A
 Fila: B, C, D
 Fila: C, D, E
 Fila: D, E, F, G
 Fila: E, F, G, H
 Fila: F, G, H

Fila: G, H
 Fila: H
 Fila: I, J, K
 Fila: J, K
 Fila: K
 Fila: vazia

```

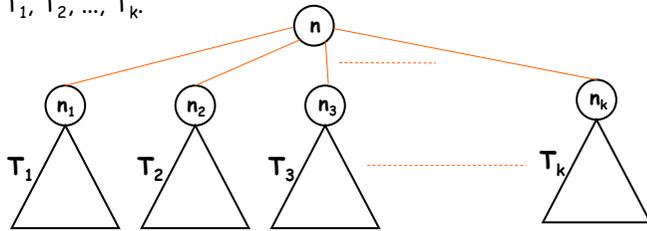
void OrdenacaoPorNivel(raiz)
{
  Fila q;
  node p = raiz;
  if (p != null) {
    enqueue(q,p);
    while (!isEmpty(q)){
      p = first(q);
      dequeue(q);
      escreve(p);
      Para todo f filho de p
        enqueue(q, f);
    }
  }
}
  
```

Resultado:

A B C D E F G H I J K

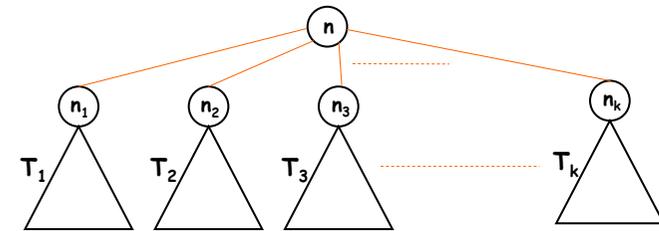
Percursos em profundidade

- Se T é uma árvore nula, então uma lista vazia será o percurso em pré-ordem, pós-ordem e ordem-central de T .
- Se T consiste em um só nó, esse nó será o percurso em pré-ordem, pós-ordem e ordem-central de T .
- Outros casos: seja T uma árvore de raiz n e sub-árvores T_1, T_2, \dots, T_k .



Percurso em pré-ordem

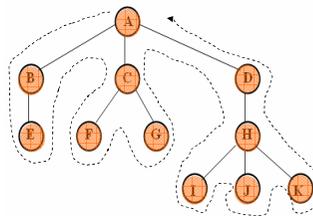
- Passos:
 - Primeiramente, a raiz n de T ;
 - Em seguida, os nós de T_1 em pré-ordem;
 - Depois, os nós de T_2 em pré-ordem;
 - Assim por diante, até os nós de T_k em pré-ordem.



Percurso em pré-ordem

- Artifício manual: anotar o nó ao passar por ele pela primeira vez

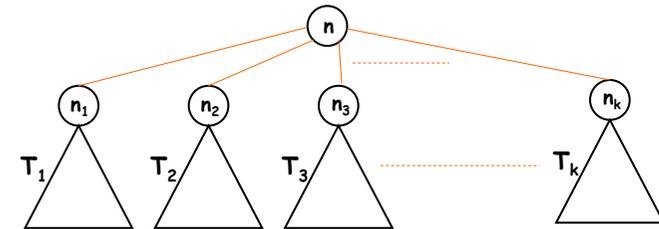
A B E C F G D H I J K



```
void PreOrdem (node n) {  
    Escrever (n);  
    for (cada filho f de n, da esquerda para a direita)  
        PreOrdem (f);  
}
```

Percurso em pós-ordem

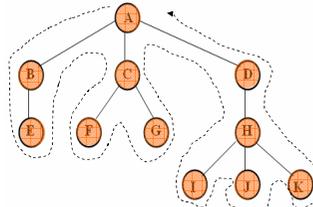
- Passos:
 - Primeiramente, os nós de T_1 em pós-ordem;
 - Em seguida, os nós de T_2 em pós-ordem;
 - Assim por diante, até os nós de T_k em pós-ordem;
 - Por fim, a raiz n de T .



Percurso em pós-ordem

- Artifício manual: anotar o nó ao passar por ele pela última vez, ou seja, ao se dirigir para seu pai.

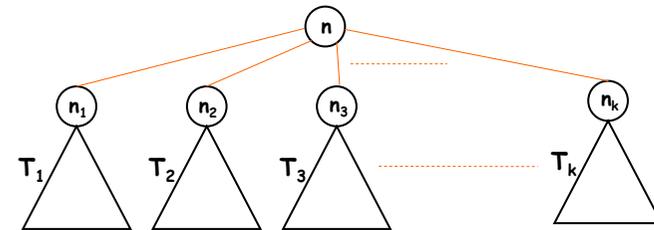
E B F G C I J K H D A



```
void PosOrdem (node n) {  
  for (cada filho f de n, da esquerda para a direita)  
    PosOrdem (f);  
  Escrever (n);  
}
```

Percurso em ordem-central

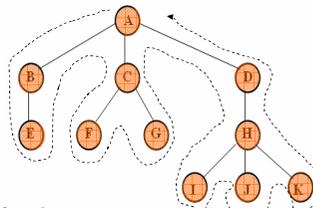
- Passos:
 - Primeiramente, os nós de T_1 em ordem-central;
 - Em seguida, a raiz n de T ;
 - Por fim, os nós de T_2, \dots, T_k em ordem-central.



Percurso em ordem-central

- Artifício manual: anotar um nó folha ao passar por ele pela primeira vez e um nó não-terminal ao passar por ele pela segunda vez.

E B A F C G I H J K D



```
void OrdCentral (node n) {  
  if (n é folha) Escrever (n);  
  else {  
    OrdCentral (Filho esquerdo de n);  
    Escrever (n);  
    for (cada filho c de n, exceto o esquerdo, da  
         esquerda para a direita)  
      OrdCentral (c);  
  }  
}
```

Fim