Estruturas de dados

Sistemas para Internet
UFSM

Árvores binárias:
conceitos e construção

Leandro O. Freitas
leandro@politecnico.ufsm.br

Retomando...

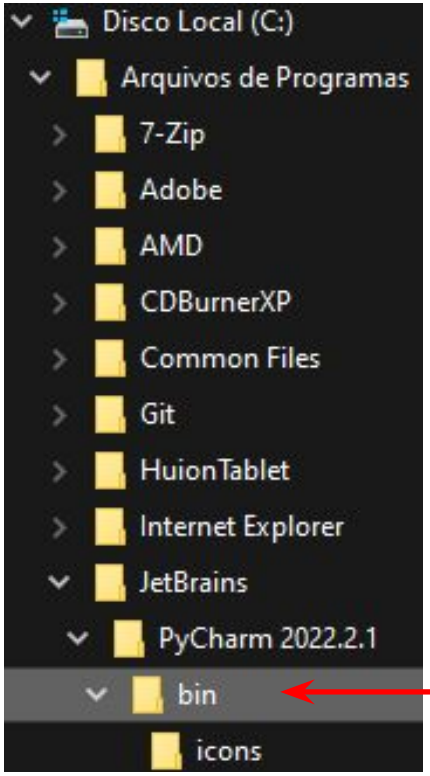
Sistemas para Internet
UFSM

- Pilhas e filas;
- Recursividade.

Árvores

Sistemas para Internet
UFSM

- Estrutura de dados utilizada para representação de dados com hierarquia.
  - Exemplos: árvore genealógica, estrutura de arquivos e pastas em um computador.
  - Diretório base, com arquivos, pastas e subpastas, que por sua vez podem ter outras subpastas e arquivos.

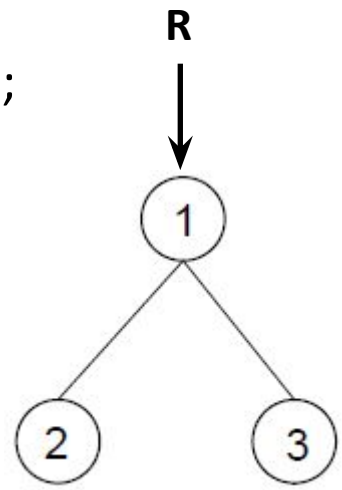


Só é possível chegar até a pasta bin, acessando primeiro suas superiores.

Árvores

Sistemas para Internet
UFSM

- Compostas por um conjunto de *nós*, onde*:*
  - Um nó R representa a raiz e pode ter zero ou mais sub-árvores;
  - As raízes das sub-árvores devem estar ligadas a raiz.



- Nomenclatura:
  - Subárvore: subdivisão da árvore em nós raízes com zero ou mais filhos;
  - Pai: nós que possuem sub-árvores;
  - Filhos (nós internos): nós das sub-árvores;
  - Folhas: nós que não possuem filhos.

- Podemos utilizar funções recursivas para representação de árvores.

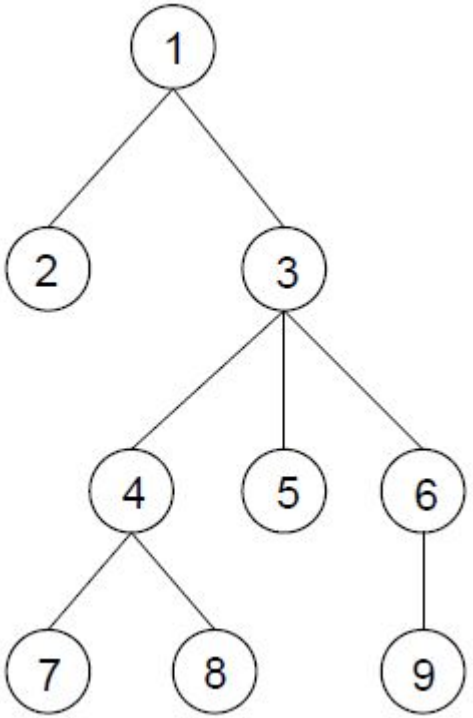
Representação

Sistemas para Internet
UFSM

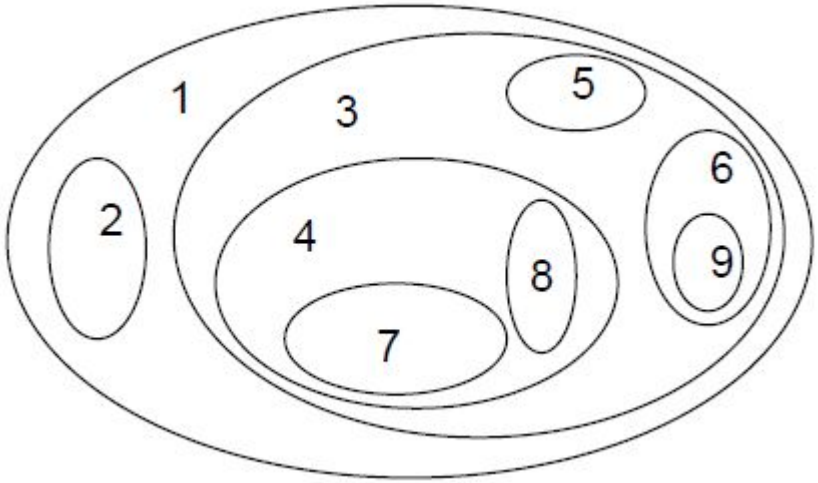
- Parênteses aninhados

(1 (2) (3 (4 (7) (8)) (5) (6 (9))))

- Hierarquia



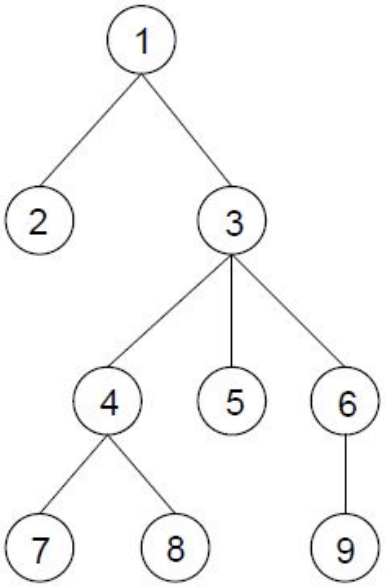
- Diagrama de inclusão



Definição de árvores

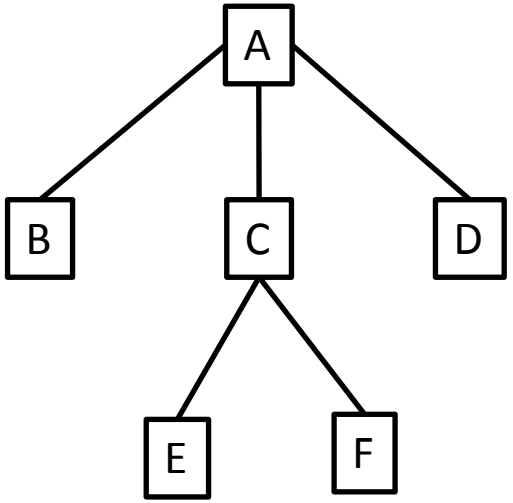
Sistemas para Internet
UFSM

- **Raiz:** nó de origem da árvore.
- **Folhas:** nós que não têm filhos.
- **Grau de um nó:** é o número de subárvores (filhos) de um nó.
- **Nível de um nó:** número de nós no caminho da raiz até um nó.
- **Altura da árvore:** é o nível mais alto da árvore.
- **Sub-árvore:** todo nó contido na árvore é a raiz de uma sub-árvore.
- **Caminho:** é o caminho entre dois nós de um árvore.
- **Árvore cheia ou completa:** uma árvore que possui o número máximo de nós, isto é, todos os nós têm número máximo de filhos exceto as folhas, e todas as folhas estão na mesma altura.



Exemplo

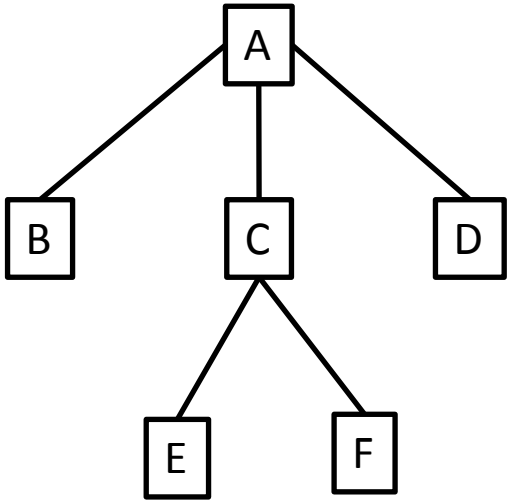
- Quais são os nós folhas:
- Quais são nós filhos?
- Quais são nós pais?



- Número de filhos em cada nó e as informações armazenadas neles variam de acordo com o tipo de árvore utilizado.

Exemplo

- Quais são os nós folhas:
  - B, D, E e F.

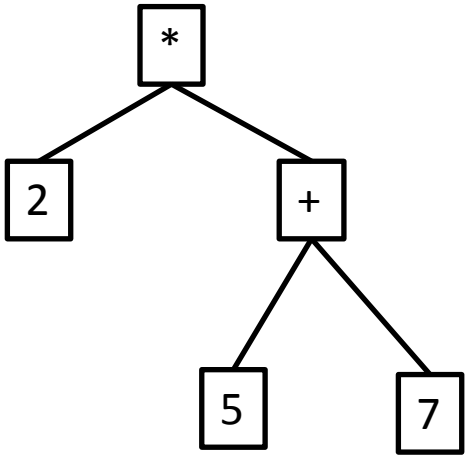


- Quais são nós filhos?
  - B, C, D com relação a A.
  - E, F com relação a C.
- Quais são nós pais?
  - A com relação a B, C e D.
  - C com relação a E e F.
- Número de filhos em cada nó e as informações armazenadas neles variam de acordo com o tipo de árvore utilizado.

Árvores binárias

Sistemas para Internet
UFSM

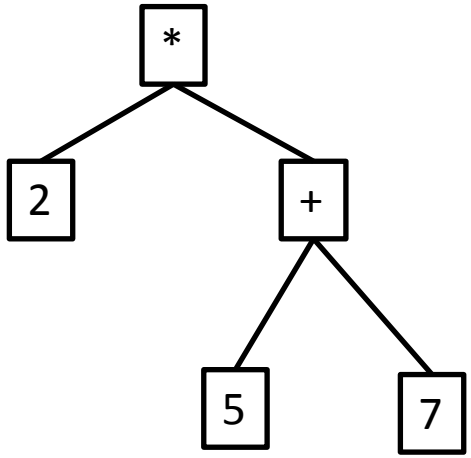
- Cada nó pode ter no máximo dois filhos (folhas ou nós internos).
- Bastante utilizada na área da matemática:
  - Folhas representam os números;
  - Nós internos representam as operações;
  - Qual expressão matemática definida na árvore?



Árvores binárias

Sistemas para Internet
UFSM

- Cada nó pode ter no máximo dois filhos (folhas ou nós internos).
- Bastante utilizada na área da matemática:
  - Folhas representam os números;
  - Nós internos representam as operações;
  - Qual expressão matemática definida na árvore?

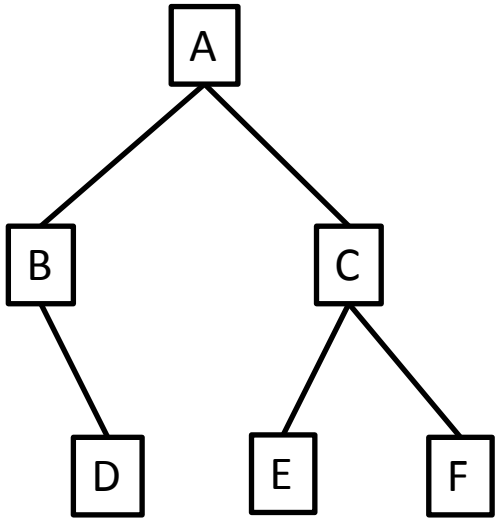


2 * (5 + 7)

Árvores binárias

Sistemas para Internet
UFSM

- Definição:
  - Vazia: sem nós;
  - Nó raiz com duas sub-árvores:
    - esquerda;
    - direita.



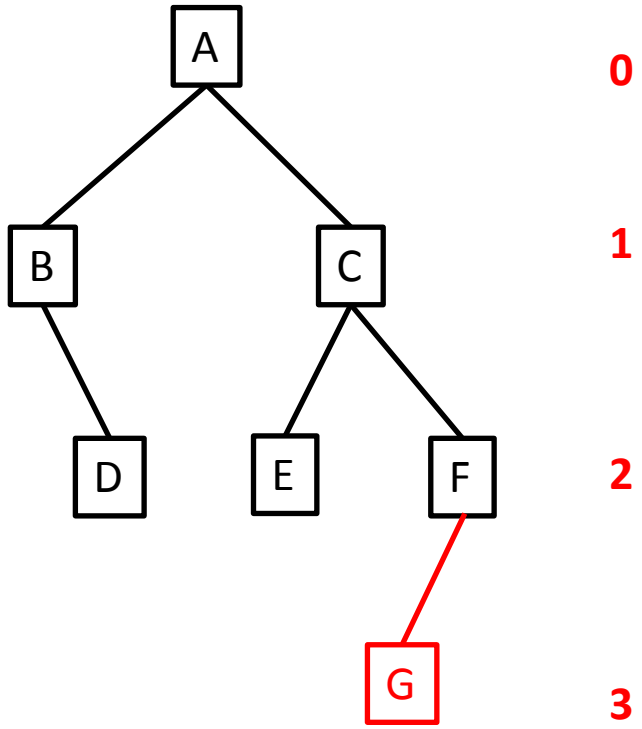
- Altura da árvore: nível do nó folha mais distante da raiz.
  - Altura da árvore do exemplo é 2.

- Árvore estritamente binária: quando todos os nós possuem 2 filhos, com exceção dos nós folha.
  - ATENÇÃO: Árvore estritamente binária != de árvore binária cheia.

Árvores binárias

Sistemas para Internet
UFSM

- Definição:
  - Vazia: sem nós;
  - Nó raiz com duas sub-árvores:
    - esquerda;
    - direita.



- Altura da árvore: nível do nó folha mais distante da raiz.
  - Altura da árvore do exemplo é 2.
- Árvore estritamente binária: quando todos os nós possuem 2 filhos, com exceção dos nós folha.
  - ATENÇÃO: Árvore estritamente binária != de árvore binária cheia.

Árvore de busca binária

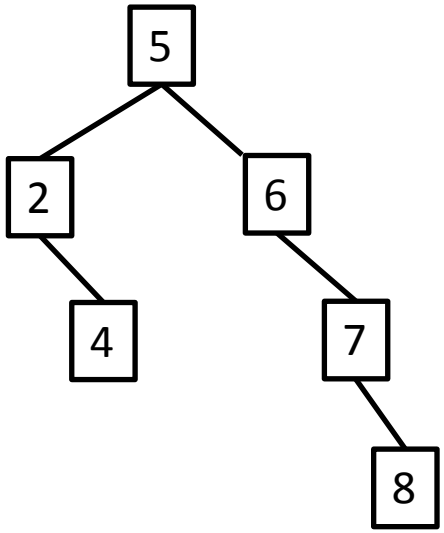
Sistemas para Internet
UFSM

- Possuem nós com a seguintes características:
  - Filho da esquerda: valor menor ou igual a raiz de sua sub-árvore;
  - Filho da direita: valor maior que a raiz de sua sub-árvore;
  - Exemplo: criar uma árvore com os elementos 2, 4, 5, 6, 7 e 8, assumindo o valor 5 armazenado na raiz.

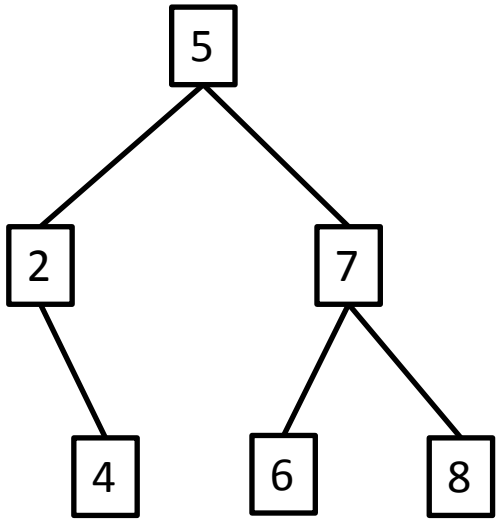
Árvore de busca binária

Sistemas para Internet
UFSM

- Possuem nós com a seguintes características:
  - Filho da esquerda: valor menor ou igual a raiz de sua sub-árvore;
  - Filho da direita: valor maior que a raiz de sua sub-árvore;
  - Exemplo: criar uma árvore com os elementos 2, 4, 5, 6, 7 e 8, assumindo o valor 5 armazenado na raiz.

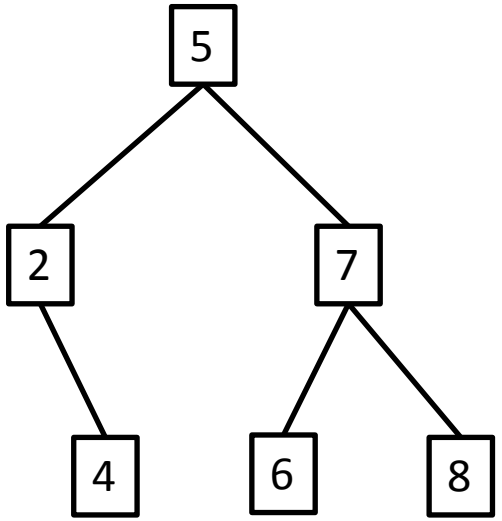


A ordem de inserção dos elementos na árvore pode influenciar em sua estrutura!!!



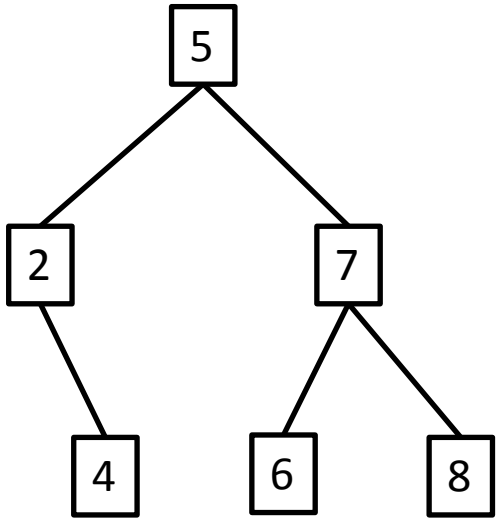
Árvore de busca binária

- Principal vantagem: facilidade de localização de um valor qualquer a partir da raiz. Exemplo:
  - Buscar o valor 8:
    - A busca é iniciada analisando a raiz: 5;



Árvore de busca binária

- Principal vantagem: facilidade de localização de um valor qualquer a partir da raiz. Exemplo:
  - Buscar o valor 8:
    - A busca é iniciada analisando a raiz: 5;
    - 8 > 5: segue pela direita;
    - 8 > 7: segue pela direita;
    - 8 encontrado.

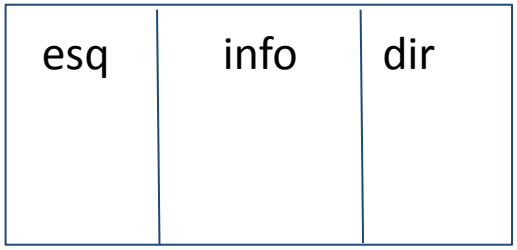


Árvores binárias

Sistemas para Internet
UFSM

- Estrutura em C:
  - Semelhante a lista encadeada, porém, ao invés usar um ponteiro para o primeiro nó, teremos um ponteiro para a raiz da árvore;
  - Cada nó deve armazenar um (conjunto de) valor(es), e dois ponteiros para as subárvores da esquerda e direita.

```
struct arvore {
    char info;
    struct arvore *esq;
    struct arvore *dir;
};

typedef struct arvore arv;
```



Árvores binárias

Sistemas para Internet
UFSM

- Principais funções:
  - Inicializar árvore vazia;
  - Criar árvores não-vazias;
  - Adicionar nós;
  - Imprimir o conteúdo da árvore;
  - Busca por valor;
  - Retirar nós.

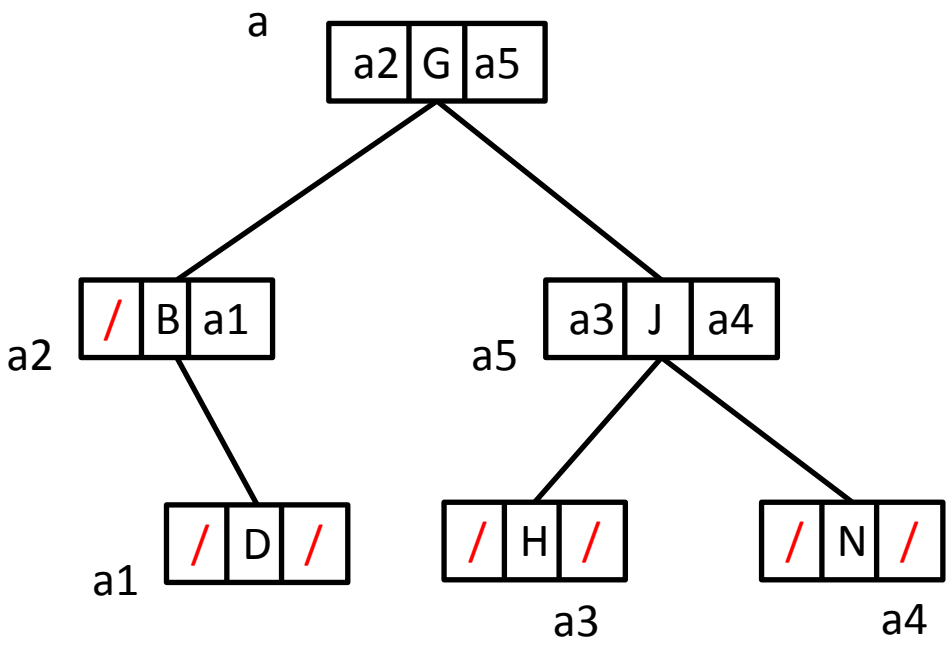
```
struct arvore {
    char info;
    struct arvore *esq;
    struct arvore *dir;
};

typedef struct arvore arv;

arv* inicializa() {
    return NULL;
}
```

Inserir elementos em posição específica

Sistemas para Internet
UFSM

- Verificar se é possível a inserção do elemento na posição desejada.
  - Deve manter as propriedades e estrutura da árvore;
  - Se posição desejada == NULL: Ok, continuar com a inserção.

É possível inserir um nó à direita de a2?
E a esquerda de a3?



Inserir elementos em árvore binária


```
13  arv *insere(arv *r, int c) {
14      arv *novo = (arv*) malloc(sizeof(arv));
15      novo->valor = c;
16      novo->esq = NULL;
17      novo->dir = NULL;
18
19      if (r == NULL)
20          r = novo;
21      else {
22          arv *aux = r, *ult;
23          while (aux !=NULL) {
24              ult = aux;
25              if (c < aux->valor)
26                  aux = aux->esq;
27              else
28                  aux = aux->dir;
29          }
30
31          if (c < ult->valor)
32              ult->esq = novo;
33          else
34              ult->dir = novo;
35      }
36      return r;
37  }
```

```
    arv *r = NULL;
0   r = insere (r, 15);
1   r = insere (r, 10);
2   r = insere (r, 8);
3   r = insere (r, 20);
4   r = insere (r, 12);
```

Realize o teste de mesa para a árvore deste exemplo.

- **aux**: para navegar pelos nós da árvore;
- **ult**: para encontrar o nó que será o pai do novo elemento.

Exercício

Sistemas para Internet
UFSM

- Adaptar o código do slide anterior para inserção dinâmica de mais elementos. A árvore deve ter altura maior ou igual a 2.