INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ

CAMPUS TERESINA-CENTRAL

DIRETORIA DE ENSINO

Estrutura de Dados II – Balanceando uma árvore - Aula 9 -

Professora: Elanne Cristina O. dos Santos

elannecristina.santos@gmail.com
elannecristina.santos@ifpi.edu.br

Por que árvores?

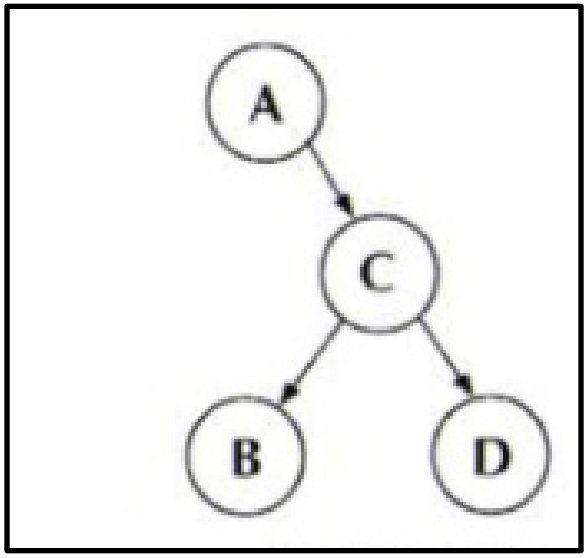
1. São bem apropriadas para representar uma estrutura hierárquica.
2. **Processo de busca muito mais rápido do que o processo em listas ligas.**

   No entanto em 2. a vantagem do processo de busca nem sempre acontece.

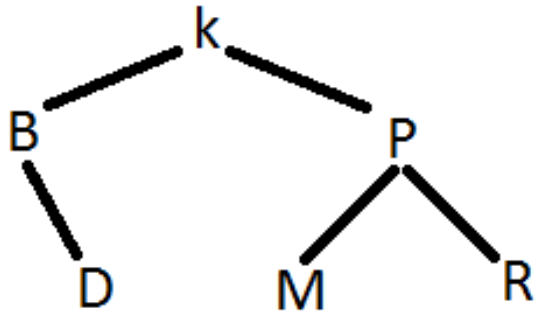
   O problema acontece quando a árvore está assimétrica (aparadas de um lado). Ou seja, quando a árvore está desbalanceada.

Árvores assimétricas ou desbalanceadas

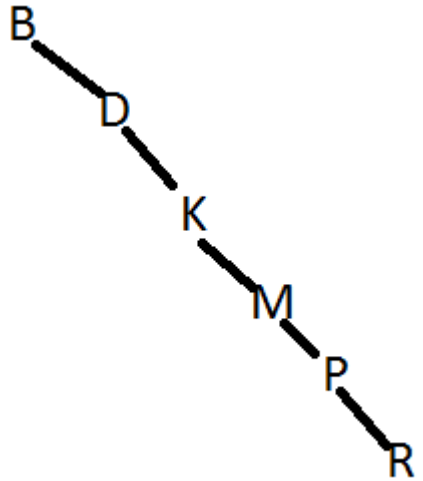
- Existe uma diferença maior que 1 entre a altura das subárvores da árvore.



Árvores assimétricas ou desbalanceadas



Diferença 1 na altura (h) entre as subárvores



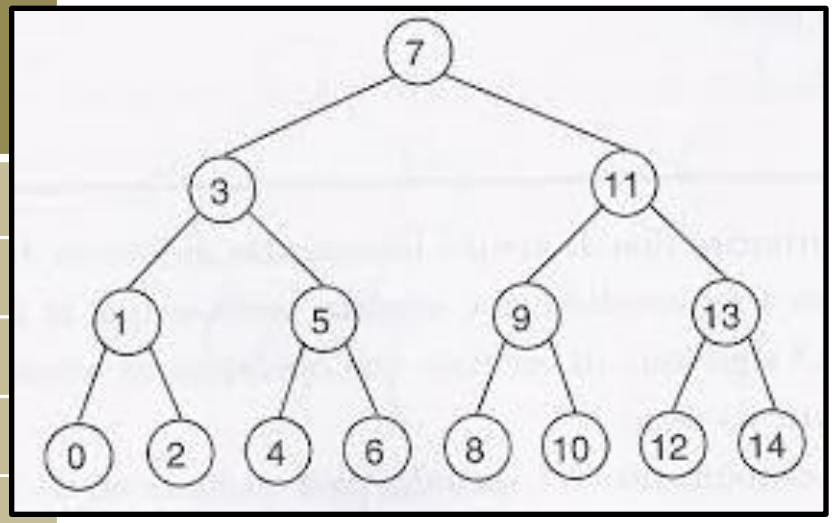
- **Diferença 6 na altura (h) entre as subárvores.**
- **TOTALMENTE ASSIMÉTRICA**
- **SEMELHANTE A UMA LISTA LIGADA**
- **PIOR CASO**

Árvore balanceada

- Uma árvore é balanceada em altura ou simplesmente balanceada se a diferença na altura de ambas as subárvores de qualquer nó na árvore é zero ou um.
- O processo de busca numa árvore balanceada é **extremamente mais eficiente** se compara a uma lista ligada, **assim é válido o esforço para construir uma árvore balanceada ou modificar uma existente de modo que seja balanceada.**

Número máximo de nós em árvores binárias de diferentes alturas

ALTURA	NÓS EM UM NÍVEL	NÓS EM TODOS OS NÍVEIS
1	$2^0 = 1$	$1= 2^1 - 1$
2	$2^1 = 2$	$3= 2^2 - 1$
3	$2^2 = 4$	$7= 2^3 - 1$
4	$2^3 = 8$	$15= 2^4 - 1$
..		
11	$2^{10} = 1.024$	$2.047= 2^{11} - 1$
..		
h	2^{h-1}	$n= 2^h - 1$
...		



Técnicas de balanceamento

- Há um número de técnicas para balancear uma árvore binária.
- Algumas consistem em reestruturar constantemente a árvore quando os elementos são inseridos e levam a uma árvore desbalanceada.
- Algumas consistem em reordenar os próprios dados e então construir a árvore balanceada.

Técnicas de balanceamento – reordenando os dados

- Armazene inicialmente os dados em um vetor.
- Se todos os dados chegaram ordene o vetor utilizando um método de ordenação *(sort)*.
- Designe, após a ordenação, para a raiz da árvore o elemento do meio do vetor.
- O vetor agora consiste em 2 subvetores:
  - uma entre o seu ínicio e o elemento raiz;
  - Outra a partir da raiz e a extremidade final do vetor.

Técnicas de balanceamento – reordenando os dados

- Primeiro a raiz é inserida em uma árvore inicialmente vazia, depois seu filho a esquerda, então seu filho a direita e assim por diante.

- **Implementação usando recursão:** primeiro insere-se a raiz (vetor[meio]), depois seu filho à esquerda(primeiro a (meio-1)), então o filho à esquerda deste à esquerda, e assim por diante. Logo em seguida o algoritmo faz a mesmo processo à direita ((meio+1) ao último).

Técnicas de balanceamento – reordenando os dados

```
template<class T>
void balancear(T vetor[], int first, int last){
     if (first<=last) {
          int middle = (first + last)/2;
          insert(vetor[middle]);
          balancear(vetor,first,middle-1);
          balancear(vetor,middle+1,last);
     }
}
```

OBS: Antes de aplicar o algoritmo "balancear" ordene o vetor com algum método *sort (bubble sort*, por exemplo).

Atividade

1) Inclua os seguintes valores na seguinte ordem em uma árvore binária:

7, 6, 22, 14, 40, 63.

1.1) Qual a altura da árvore resultante?

1.2) Mostre a árvore resultante.

1.3) A árvore resultante está balanceada ou não?

1.4) No caso da árvore resultante NÃO ESTAR balanceada aplique o algoritmo de reordenação dos dados apresentado anteriormente para balancear a árvore.