

# Estrutura de Dados II

Prof. Me. Pietro M. de Oliveira

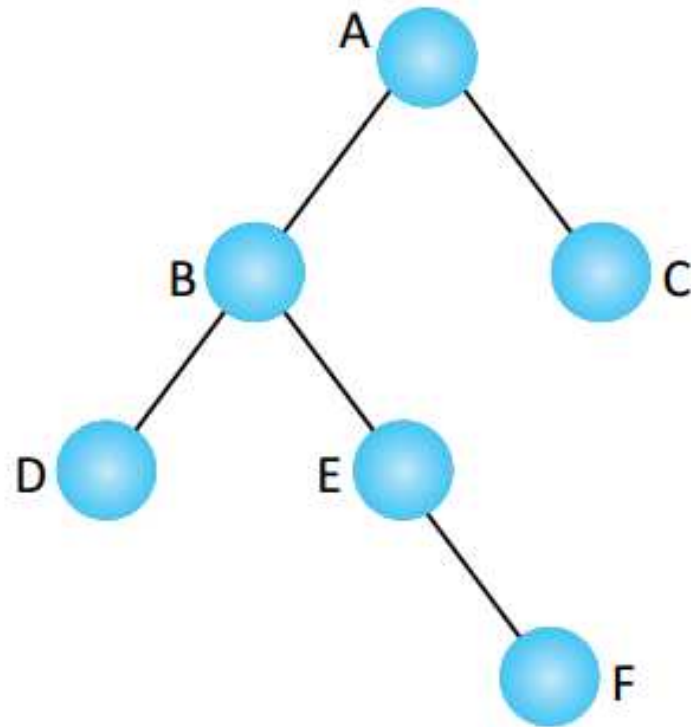
# Implementação de Árvores Binárias

- Implementação:
  - Estática – sem registros, com registros
  - Dinâmica – utiliza registros com ponteiros
- Busca em árvores binárias
  - Árvore binária ordenada
  - Árvore balanceada
    - Nós distribuídos uniformemente nos níveis
    - Tentativa de tornar “binária completa”
    - Algoritmo de balanceamento
      - Árvore AVL

- Árvore binária estática
  - Operação de adição de nó simples torna o vetor “bagunçado”.
  - Utiliza apenas um vetor de inteiros
    - Não necessita de estrutura nó (registro em C)
  - Solução: fórmula para determinar o índice dos nós (considerando posição inicial igual a 0)
    - $F_e = 2 * P + 1$
    - $F_d = 2 * P + 2$

# Árvores Binárias em Vetor de Inteiros

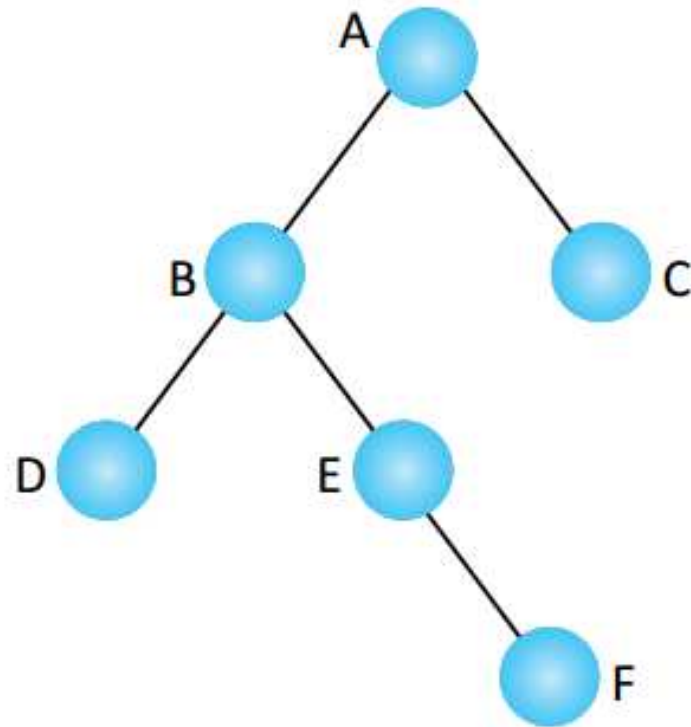
Exemplo



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A															

# Árvores Binárias em Vetor de Inteiros

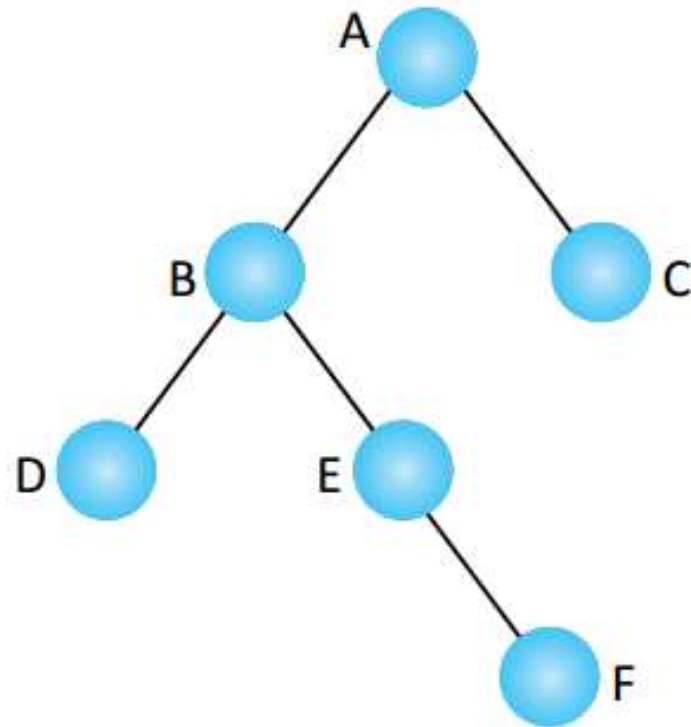
Exemplo



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B														

# Árvores Binárias em Vetor de Inteiros

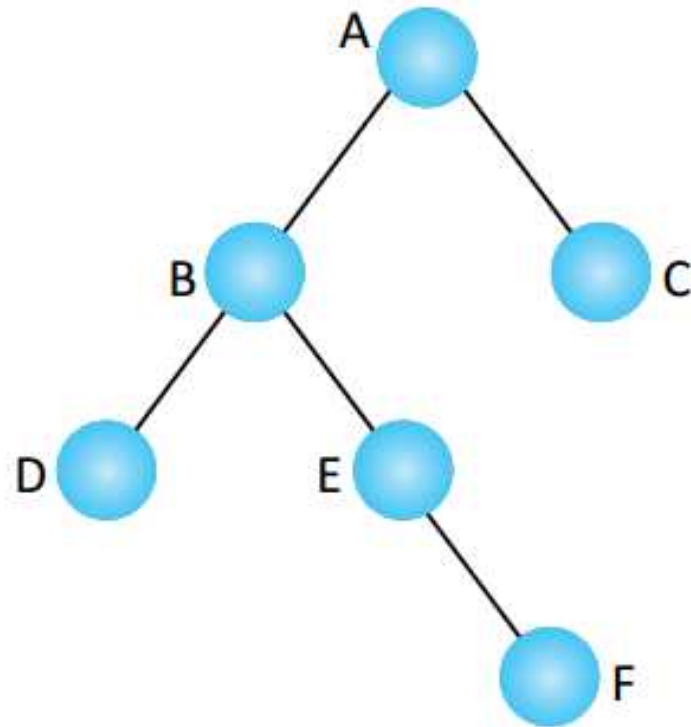
Exemplo



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C													

# Árvores Binárias em Vetor de Inteiros

Exemplo

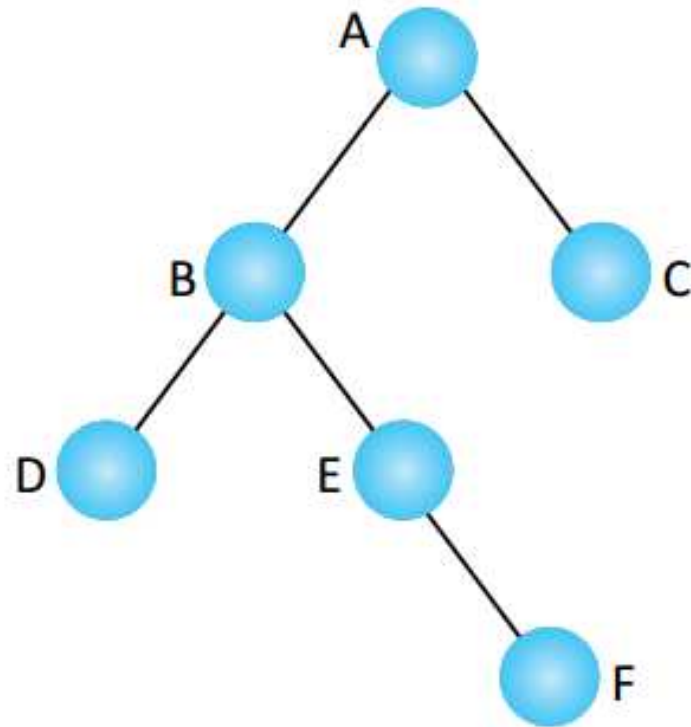


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D												



# Árvores Binárias em Vetor de Inteiros

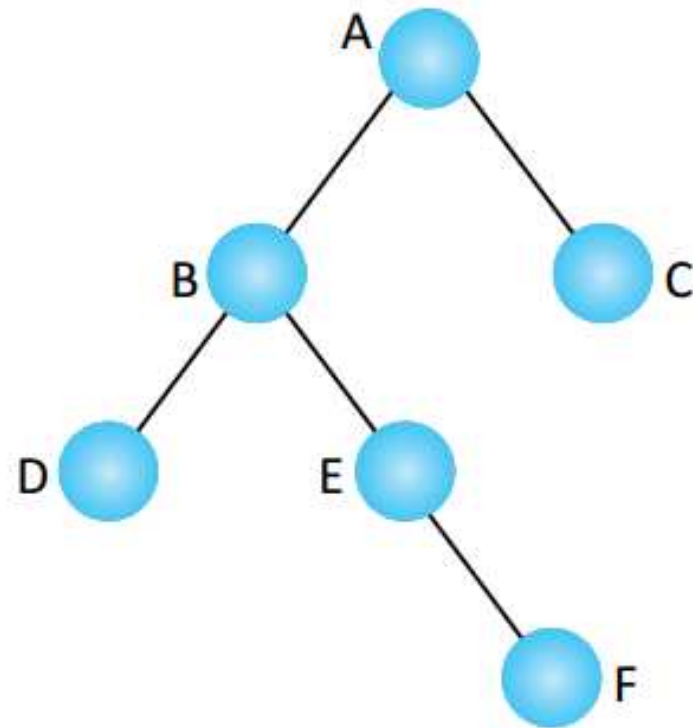
Exemplo



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E											

# Árvores Binárias em Vetor de Inteiros

Exemplo



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E						F					

E se fossemos inserir um filho à esquerda de D?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D	E			X			F					

Imagine que a posição inicial do vetor tem índice 1

- $F_e = 2 * P$
- $F_d = 2 * P + 1$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	B	C	D	E						F					