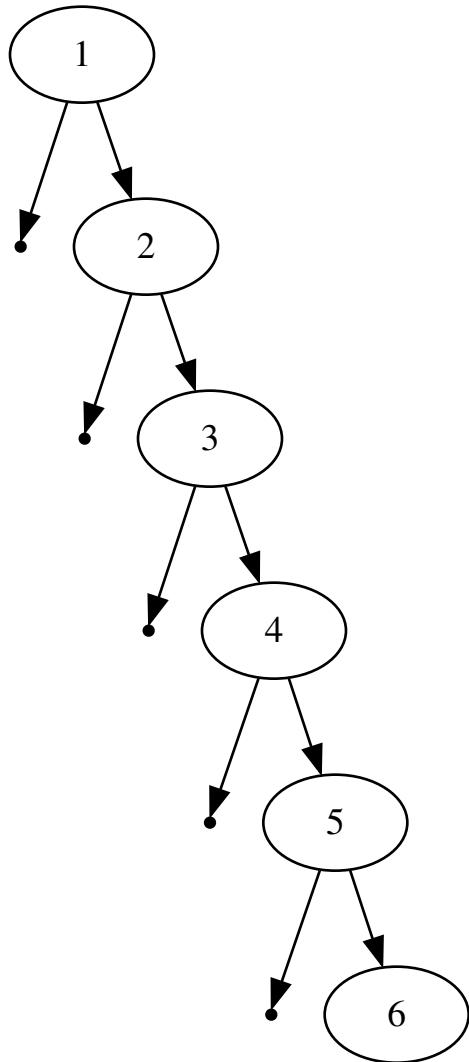


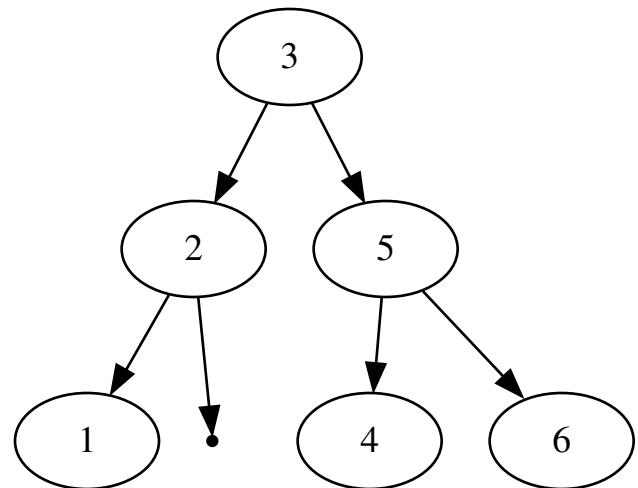
# **Balanceamento de ABBs**

**Algoritmos e Estruturas de Dados**

A ordem de inserção afeta diretamente a altura de nossas árvores!



A ordem de inserção afeta diretamente a altura de nossas árvores!



# Balanceamento

- A altura  $h$ , portanto, pode no pior caso ser  $\mathcal{O}(N)$ .
- O desempenho médio não é ruim:  $E(h) = \mathcal{O} \log N$
- Seria uma boa conseguir "consertar" uma árvore ruim!

**Definição:** uma (sub)árvore com  $N$  elementos está balanceada se  $h = \log N$ .

# Alguma primeira ideia de como balancear uma árvore?

- **Dica 1:** pode usar array auxiliar
- **Dica 2:** pode criar uma nova árvore

# Balanceamento (I)

## Algoritmo:

1. passa por todos elementos em-ordem, guardando o valor em um array
2. insere como raiz o elemento do meio do array
3. faz o passo 2 recursivamente para cada lado do array

## Complexidade:

- Computacional?
- Espaço?

# **Balanceamento (II)**

**Isso é bom?**

**Extra: procure pelo algoritmo Day-Stout-Warren**

# Árvores balanceadas

Algoritmos de inserção/remoção que mantém  $h = \log N$

- AVL
- Red-Black Tree

**Não veremos nesta disciplina**