

Collab

Métodos de Pesquisa e Ordenação em Estruturas
de Dados

Introdução

- Estrutura de dados é o nome dado à organização de dados e algoritmos de forma coerente e racional, de modo a otimizar seu uso;
- Responsável por manter os dados organizados, seguindo alguma lógica, e disponibilizar operações para o usuário manipular esses dados;
- Tipos simples são aqueles comumente utilizados;
- Atribuição de valores: string com " " ou ' '. Números, TRUE, False, etc, respeitar os tipos → `a=2` ou `nome="Fellipe"`;
- Tipos compostos - Formados pela combinação ou diversos dados de um tipo primitivo → `b=[1,2,3,4]`

Tipos Compostos

- Tupla é uma lista imutável - tipo de dados que não é alterado, é possível utilizar a tupla para dados constantes e estáticos
 - É possível utilizar uma lista e converter ele numa nova variável tupla usando a função `tuple(variavel)`
- set - A manipulação dos dados é feita pelo próprio python, elementos repetidos são consolidados num set,
 - Uso parecido com uma lista, a organização fica a cargo da estrutura
- Dicionário: relação entre valores, necessário uma chave relacionando o índice aos dados

Notação big O

- A eficiência dos tempos de execução é dada pela sequência $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$;
- A notação O é utilizada, para exprimir a complexidade de determinado algoritmo;
- Uma das ideias seria determinar o pior, melhor e caso médio de determinado algoritmo, ou simplesmente como ocorre o crescimento do tempo de execução;

Lista

- Semelhante à uma lista de supermercado ou de convidados, onde os itens são dispostos em sequência de valores ordenados
- São do mesmo tipo, sendo a representação de uma sequência de objetos
- Métodos `append()` e `pop()` - remove (exclui quando sabe o valor)

Fila

- Nas filas, o primeiro elemento a ser adicionado é sempre o primeiro a ser removido
- Trata-se de um tipo de dados do tipo FIFO (First-in – First-out)
- Métodos `append()` - insere e `pop()` remove()
- Também é possível atribuir um valor específico
 - `a[0]=c`

Pilha

- Pilha – Estrutura que se assemelha a forma de organizar livros ou pratos (conforme exercício);
- O primeiro item adicionado a uma pilhas é o último a ser retirado de uma pilha;
 - Usado no sistema operacional para salvar o contexto de uma tarefa e num *browser* como os botoes voltar
- Os últimos itens adicionados são os primeiros a serem removidos – LIFO;
- Métodos `append()` - insere e `pop()` remove do final da estrutura
- Pode ser usado `popleft()` - remove do início da pilha – importar a estrutura `deque` e declarar a variável como `deque()`

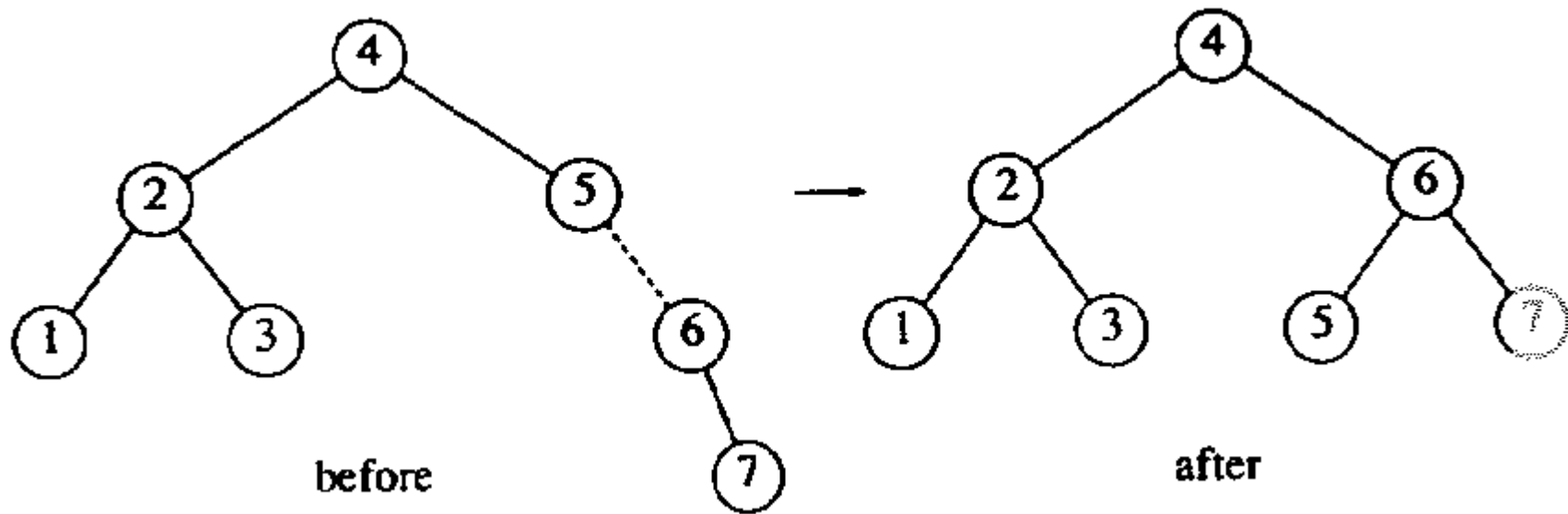
Métodos de Ordenação

- InsertionSort - Cada elemento novo é comparado com os elementos do vetor, ele verifica cada item com todos os outros já ordenados, guarda posição - esquerda
 - O melhor caso do é a situação em que os dados já estão ordenados em ordem crescente - $O(n)$
- BubbleSort – Comparar todo o vetor várias vezes e invertendo os dados quando o número comparado for maior do que a próxima posição do vetor - direita.
 - Não é muito eficiente quando tratamos de uma grande quantidade de dados devido a quantidade de trocas que são realizadas
- QuickSort. - divisão, conquista e combinação. O QuickSort divide o problema em pedaços menores e cada pedaço é resolvido individualmente, posteriormente esses dados são reunidos novamente.
 - Situação crítica é quando o pivô está no final ou no início do vetor, melhor caso é quando o pivô está no meio
- <https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

Árvores Hierárquicas

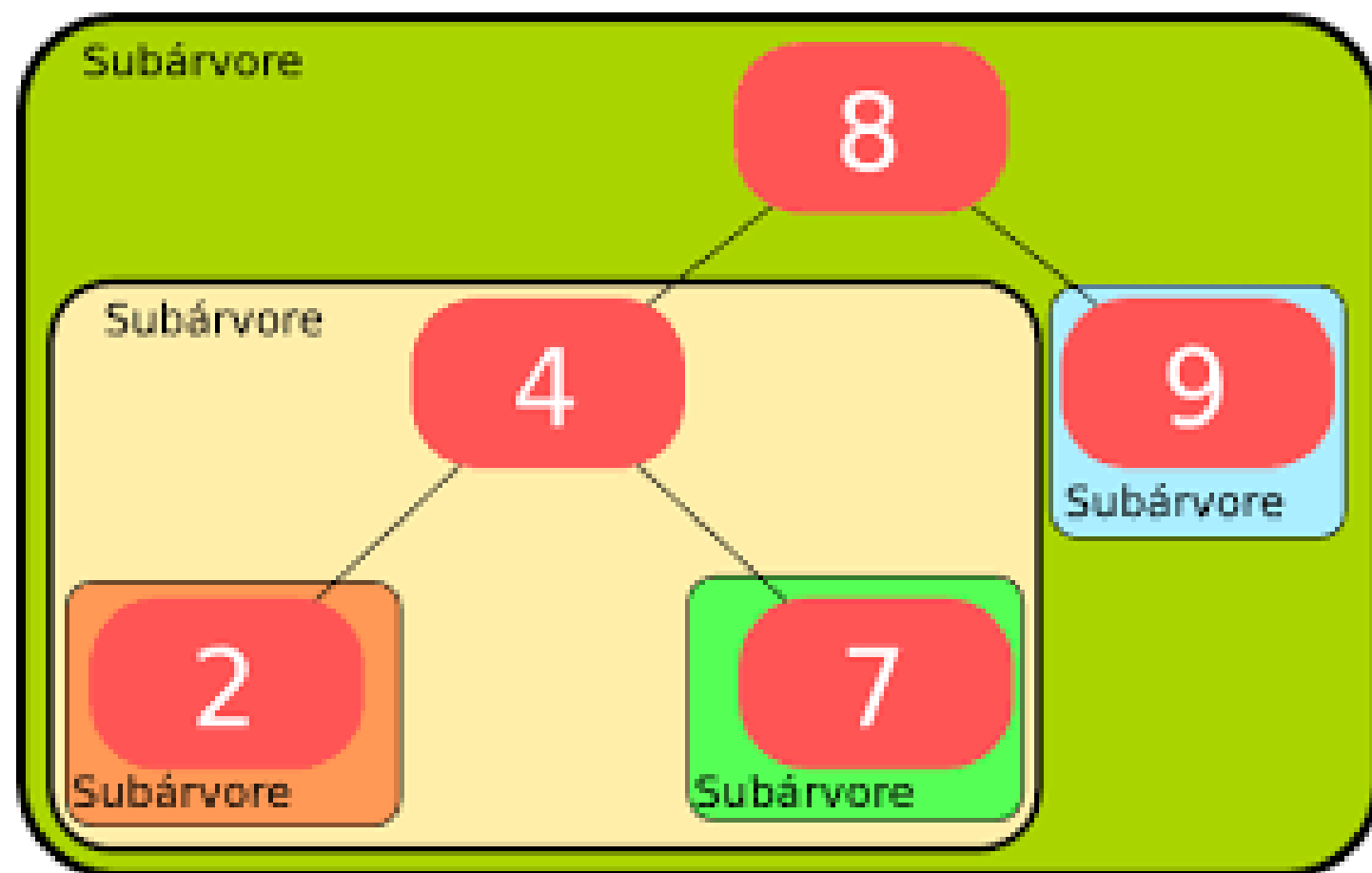
- São binárias, portanto cada nó da árvore pode possuir 0,1 ou no máximo 2 filhos
- Se o nó não possui filho é chamado de folha ou nó-folha
- Tornar mais eficiente a tarefa de procura dos elementos
 - Uma árvore normal pode ter um lado com mais folhas e mais profunda, tornando a busca mais lenta
 - O elemento deve ser procurado à esquerda (menor que o item atual), ou à direita (maior do que o item atual)
 - Para isso a árvore deve estar balanceada
 - Balanceamento estático: Ocorre uma vez
 - Balanceamento dinâmico: Rotações ocorrem quando acontece uma nova entrada

Comparativo



Árvore

- A altura de uma árvore binária seria o nível máximo das suas folhas, também chamada de profundidade
- Se a probabilidade de pesquisar um dado for a mesma para todos os dados, uma árvore binária balanceada determinará a busca mais eficiente



Métodos de Pesquisa

- Pesquisa sequencial: Linear, é um dos métodos mais simples entre aqueles encontrados na literatura. Percorrer a lista procurando um valor idêntico ao buscado.
- Pesquisa binária: Divisão da lista em sublistas menores. O principal problema desse tipo de pesquisa é a necessidade de manter a lista de entrada ordenada
- Pesquisa hash: Mapear dados de tamanho variável ou grandes em dados pequenos de tamanho fixo possibilita a resolução ou simplificação de diversos problemas
 - De acordo com forma escolhida para transformação dos dados, é possível minimizar a ocorrência de colisões (informações distintas com mesmo hash)
 - Necessita comparar o valor buscado com todos os elementos do vetor,

Pesquisa Binária

- Caso o vetor esteja desordenado, o primeiro passo é realizar a ordenação
- Identifique a quantidade de valores existentes no problema, esse valor deve ser um inteiro e deve ser armazenado na variável max. A variável min deve conter 0.
- Dividir o valor de (max-min) por dois. Caso o número não seja um inteiro arredonde para baixo.
- Compare o valor pesquisado com o valor encontrado na posição calculada na etapa anterior.
- Se o valor for menor que o valor pesquisado, a variável min recebe o valor calculado + 1. Se for maior, a variável max recebe o valor calculado -1.
- Caso seja igual o valor foi encontrado.
- Com os novos valores de min e max volte para a etapa de cálculo da posição central
- Caso as possibilidades se acabem (min == max) o valor não está presente no vetor