

• TREES

Uma árvore (mais precisamente, uma árvore livre) é um grafo acíclico conectado. Um grafo que não possui ciclos, mas não é necessariamente conectado, é chamado de floresta: cada de seus componentes conectados é uma árvore. As árvores têm várias propriedades importantes que outros gráficos não têm. Em particular, o número de arestas em uma árvore é sempre um a menos que o número de seus vértices. Outra propriedade muito importante das árvores é o fato de que para cada dois vértices em uma árvore, sempre existe exatamente um caminho simples de um desses vértices para o outro. Esta propriedade torna possível selecionar um vértice arbitrário em uma árvore livre e considerá-la como a raiz da chamada árvore enraizada. Para qualquer vértice v em uma árvore T , todos os vértices no caminho simples da raiz a esse vértice são chamados ancestrais de v . Se (u, v) é a última aresta do caminho simples da raiz para o vértice v ($u = v$), u é considerado o pai de v e v é chamado de filhos. Um vértice sem as crianças são chamadas de folhas. Todos vértices para os quais um vértice v é um ancestral são ditos descendentes de v ; o descendentes próprios excluem o próprio vértice v . A profundidade de um vértice v é o comprimento do caminho simples da raiz até v . A altura de uma árvore é o comprimento do caminho simples mais longo desde a raiz até uma folha.

• BINARY TREES

Uma árvore binária pode ser definida como uma árvore ordenada na qual cada vértice tem não mais do que dois filhos e cada filho é designado como filho esquerdo ou filho direito de seu pai; uma árvore binária também pode estar vazia. A árvore binária com sua raiz no filho esquerdo (direito) de um vértice em uma árvore binária é chamada de subárvore esquerda (direita) desse vértice. Desde as subárvores esquerda e direita também são árvores binárias, uma árvore binária também pode ser definida recursivamente. Isso torna possível resolver muitos problemas envolvendo árvores binárias por algoritmo recursivo.

TREES

• TRAVESSIAS

No percurso de pré-ordem, a raiz é visitada antes das subárvores esquerda e direita são visitados (nessa ordem).
Na travessia inorder, a raiz é visitada depois de visitar sua subárvore esquerda, mas antes de visitar a subárvore direita.
Na travessia de pós-ordem, a raiz é visitada depois de visitar a esquerda e a direita subárvores (nessa ordem).

• BINARY SEARCH TREES

Observe que um número atribuído a cada vértice parental é maior do que todos os números em sua subárvore esquerda e menor que todos os números em sua subárvore direita. Essas árvores são chamadas de árvores binárias de busca. Árvores binárias e árvores binárias de busca têm uma ampla variedade de aplicações em ciência da computação; você vai encontrar alguns dos eles ao longo do livro. Em particular, as árvores de busca binária podem ser generalizadas a tipos mais gerais de árvores de busca chamadas árvores de busca multiway, que são indispensável para o acesso eficiente a conjuntos de dados muito grandes.

• SEARCH AND INSERTION ON BSTS

Quando precisamos procurar um elemento de um determinado valor v em tal árvore, fazemos isso recursivamente no seguinte maneira. Se a árvore estiver vazia, a busca termina em falha. Se a árvore não vazia, comparamos v com a raiz $K(r)$ da árvore. Se eles combinarem, um elemento desejado é encontrado e a busca pode ser interrompida; se não corresponderem, continuamos com a busca na subárvore esquerda da raiz se $v < K(r)$ e na subárvore direita se $v > K(r)$. Assim, a cada iteração do algoritmo, o problema de busca em um a árvore de pesquisa binária é reduzida à pesquisa em uma árvore de pesquisa binária menor. Desde a inserção de uma nova chave em uma pesquisa binária árvore é quase idêntica àquela de busca lá, ela também exemplifica a técnica de diminuição de tamanho variável e tem as mesmas características de eficiência da busca operacional