

Python Estrutura de Dados

Módulo 07
Aula 13

Estrutura de Dados

Resumo do Módulo

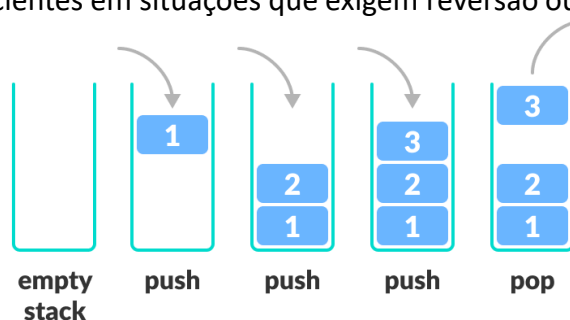
Filas

Filas são estruturas de dados que seguem o princípio FIFO (First In, First Out), onde o primeiro elemento adicionado é o primeiro a ser removido. Essencial em cenários como o gerenciamento de tarefas, operações de IO e simulações de eventos do mundo real, filas facilitam o processamento ordenado e sequencial de elementos, assegurando que as operações ocorram na ordem correta, como em sistemas de atendimento ao cliente ou na gestão de filas de impressão.



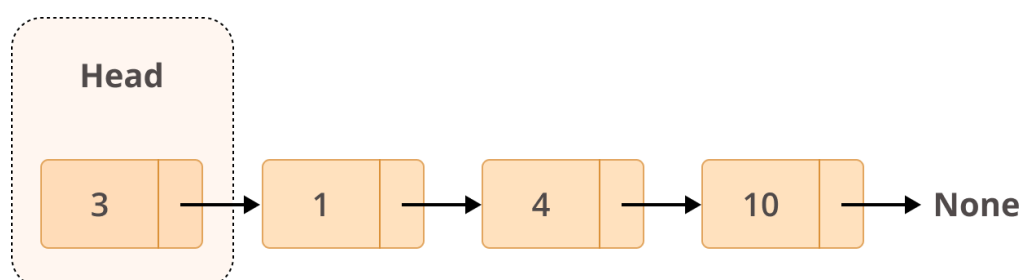
Pilhas

Pilhas operam sob o princípio LIFO (Last In, First Out), significando que o último elemento adicionado é o primeiro a ser removido. Utilizadas em uma variedade de aplicações, como a execução de chamadas de função (pilha de chamadas), análise de expressões matemáticas (notação polonesa reversa) e navegação histórica em navegadores web, as pilhas são cruciais para criar algoritmos eficientes em situações que exigem reversão ou desfazer ações.



Listas Encadeadas

Listas encadeadas são coleções de elementos chamados nós, onde cada nó contém dados e um ponteiro para o próximo nó na sequência. Flexíveis e dinâmicas, permitem inserções e exclusões eficientes sem realocação de outros elementos. As variantes incluem listas simplesmente encadeadas, duplamente encadeadas e circulares, cada uma otimizando operações específicas, como navegação reversa ou acesso cíclico.

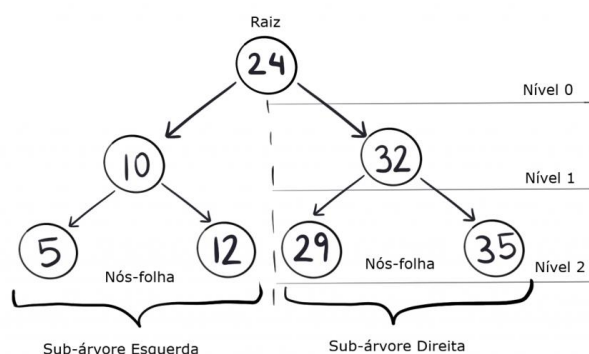


Estrutura de Dados

Resumo do Módulo

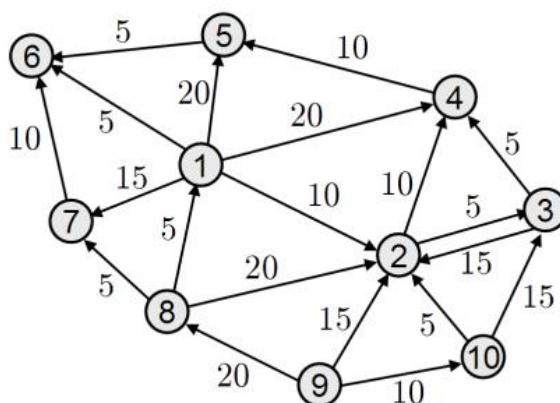
Árvores Binárias

Árvores binárias são estruturas hierárquicas com nós conectados por arestas, onde cada nó tem até dois filhos. Essenciais para representar relações hierárquicas e realizar buscas eficientes, as variantes incluem árvores binárias de busca, AVL e vermelho-preto, cada uma oferecendo soluções para manter o equilíbrio e otimizar operações de busca, inserção e remoção, fundamentais em bancos de dados e sistemas de arquivos.



Grafos

Grafos são conjuntos de vértices (ou nós) conectados por arestas, utilizados para modelar relações complexas em redes sociais, sistemas de rotas, análises de conexões e muito mais. Podem ser dirigidos ou não dirigidos, ponderados ou não, representando uma vasta gama de problemas do mundo real. Algoritmos como busca em profundidade e largura, Dijkstra e Kruskal permitem explorar grafos para encontrar caminhos, componentes conectados e árvores geradoras mínimas.



Hashmaps

Hashmaps, ou tabelas de dispersão, são estruturas de dados que mapeiam chaves únicas a valores, proporcionando inserções, buscas e remoções extremamente rápidas. Essenciais para otimizar o acesso a dados, são amplamente usados em caches, bancos de dados e para suportar operações de busca eficiente. Através de uma função de hash que distribui chaves por "buckets", os hashmaps gerenciam colisões e mantêm o desempenho mesmo com grandes volumes de dados.

Estrutura de Dados

Resumo do Módulo

Ordenação

Algoritmos de ordenação, como QuickSort, MergeSort e HeapSort, são fundamentais para organizar dados. Cada algoritmo tem suas características, como eficiência, estabilidade e complexidade, adequando-se a diferentes cenários. Por exemplo, QuickSort é rápido e eficaz para conjuntos grandes, MergeSort é estável e eficiente para dados encadeados, e HeapSort é ótimo para ordenação in-place, cada um aplicável conforme as necessidades específicas de ordenação de dados.

Recursão

Recursão é uma técnica de programação onde uma função chama a si mesma para resolver subproblemas de maneira similar. Facilita a escrita de soluções claras para problemas complexos, como a navegação em estruturas de dados hierárquicas (árvores e grafos), algoritmos de ordenação e busca, e cálculos matemáticos (fatoriais, Fibonacci). Embora poderosa, a recursão requer atenção ao caso base e ao consumo de recursos, especialmente memória de pilha.