

Programação em Python

Ideias chave

Um algoritmo é uma sequência de instruções cujo objetivo é resolver um problema.

Quando falamos sobre a complexidade de um algoritmo, estamos na verdade nos referindo a uma métrica teórica que mede a eficiência computacional de um algoritmo.

A fim de conhecer a complexidade dos algoritmos podemos classificá-los em grupos, sendo os mais comuns:

- **Constante:** $O(1)$. É a mais simples e sempre tem um tempo de execução constante.
- **Linear:** $O(x)$. O tempo cresce linearmente à medida que os dados crescem.
- **Polinomial:** $O(x^c)$, $c > 0$. Estes são os algoritmos mais comuns. Se eles tivessem 2 laços aninhados, seria um quadrático.
- **Logarítmico:** $O(\log x)$. Indicar que o tempo é menor do que o tamanho dos dados de entrada.
- **Enelogarítmico:** $O(n \log x)$. Tão bom quanto o anterior.
- **Exponencial:** $O(c^x)$. É uma das piores complexidades algorítmicas. Ele sobe muito alto à medida que os dados de entrada crescem.

Notações assintóticas são ferramentas matemáticas para representar a complexidade temporal dos algoritmos para análise assintótica. Há principalmente três notações assintóticas:

- **Notação theta (notação Θ):** encerra a função a partir de cima e de baixo.
- **Notação omega (notação Ω):** representa o limite inferior do tempo de execução de um algoritmo. Portanto, ele fornece a melhor complexidade de caso de um algoritmo.
- **Notação Big-O (notação O):** representa o limite superior no tempo de execução de um algoritmo.

Portanto, ele dá o pior cenário de complexidade de um algoritmo.

Existem quatro tipos de estruturas de coleta de dados na linguagem de programação Python:

- **A lista (list);** é uma coleção classificável e modificável. Permite a duplicação de membros.
- **Tupla (tuple);** é uma coleção ordenada e imutável. Permite a duplicação de membros.
- **Conjuntos (Set);** é uma coleção que não é classificada nem indexada. Sem duplicação de membros.
- **Dicionário (Dictionary);** é uma coleção não ordenada, modificável e indexada. Sem duplicação de membros.

Há dois tipos de algoritmos de busca:

- **Algoritmo de busca linear:** é o algoritmo de busca mais simples. Também é chamada de busca sequencial porque, neste algoritmo, procuramos um elemento atravessando todo o array e comparando cada elemento com o elemento desejado para encontrar uma correspondência. Se o elemento desejado for encontrado, o índice ou aquele elemento é devolvido; caso contrário, continuamos a busca até que o array se esgote. Também podemos procurar por múltiplas ocorrências de um elemento dentro de um array. Ele é usado principalmente para procurar elementos dentro de uma matriz não classificada. Raramente é usado porque é muito mais lento do que a busca binária.
- **Algoritmo de busca binária:** é o algoritmo de busca mais rápido. Requer que o array seja classificado. Ela se baseia na abordagem de dividir e conquistar, na qual dividimos o array em duas metades e depois comparamos o elemento que estamos procurando com o elemento do meio. Se o item do meio corresponder, devolvemos o índice do item do meio; caso contrário, nos movemos para a metade esquerda e direita de acordo com o valor do item.

Algoritmos de ordenação:

- **Algoritmo de classificação da inserção:** é um algoritmo fácil de aplicar que permite a ordenação de uma lista. Ele funciona atravessando a lista selecionando um valor como chave em cada iteração e comparando-o com o restante inserindo-o no local correspondente.
- **Ordem de seleção:** semelhante à ordem de inserção com uma ligeira diferença. Este algoritmo também divide a matriz em subpartes ordenadas e não-ordenadas.
- **Classificação de bolhas:** itera sobre a lista de dados, comparando elementos em pares até que os elementos maiores "borbulhem" até o final da lista e os menores permaneçam no topo.
- **Merge sort:** O algoritmo de separação é um algoritmo de separação externo estável baseado na técnica de dividir e conquistar. A ideia de embaralhar algoritmos de ordenação é dividir a matriz ao meio uma e outra vez até que cada peça tenha apenas um elemento de comprimento. Estes elementos são então colocados novamente juntos (embaralhados) em ordem de classificação.