

Algoritmos de Ordenação Interna

Frank Coelho de Alcantara

Ordenação

Entre os algoritmos mais importantes destacam-se os algoritmos de ordenação, em inglês *sort*, isto porque a maior parte das tarefas diárias envolve alguma forma de ordenação.

Ordenamos registros em bancos de dados, as tarefas do dia, conhecimento em parágrafos, parágrafos em páginas e páginas em sites. Poderíamos estender esta lista para lembrar que ordenamos desde muito cedo, antes mesmo de aprender a contar. Esta tarefa, a ordenação está emaranhada com o desenvolvimento da humanidade. A própria civilização, como conhecemos, deve seu início à agricultura e esta à percepção da ordem dos ciclos solares. Ao longo da história os processos de ordenação tiveram impacto na ascensão e queda de impérios e empresas.

O *Google* deve seu sucesso a uma lista ordenada criada a partir de um algoritmo de classificação, o *PageRank*. Foi este algoritmo desenvolvido por Larry Page and Sergey Brin que permitiu que o *Google* retornasse uma lista de páginas na ordem que melhor atenda as demandas de busca de um determinado usuário. Esta pequena vantagem sobre os concorrentes permitiu que o *Google* se tornasse uma empresa quase onipresente.

A missão do *Google* era, ou é, indexar todo o conhecimento da humanidade mas, seu sucesso comercial se deve a ordem que este conhecimento é apresentado.

Uma parte importante das tarefas computacionais incluem alguma forma de ordenação. Pesquisar na internet, consultar um banco de dados, ordenar uma coleção de livros, ou artigos são algumas das tarefas que os aplicativos executam sem que tenhamos percepção do que está acontecendo. Graças a isso, estes algoritmos são contínua e profundamente estudados em todas as áreas da computação. São estudados em busca de mais eficiência, para os algoritmos já conhecidos, para a criação de algoritmos novos. Este estudo é necessário porque a computação consiste na busca constante de mais eficiência e mais eficácia. Um esforço constante de otimização de algoritmos, plataformas e serviços.

No caso dos algoritmos uma das primeiras regras de otimização ressalta a importância de começar a otimização pelos algoritmos de ordenação. São estes que geralmente consomem mais tempo e recursos. Não bastasse isso, como

justificativa para o estudo dos algoritmos de ordenação, ainda é necessário destacar que esta classe de algoritmos são perfeitamente adequados tanto para os exercícios de avaliação de eficiência de algoritmos quanto para exemplo do uso das técnicas de desenvolvimento de algoritmos como: dividir-e-conquistar, estruturas de dados compostas e uso de aleatoriedade na criação de algoritmos.

Problema: dado um conjunto de dados qualquer, de que forma podemos ordenar estes dados em uma sequência determinada.

O problema da ordenação é simples, e consiste em encontrar uma forma de colocar dados em ordem. Esta ordem pode ser crescente ou decrescente e os dados podem ser de qualquer tipo, incluindo os tipos primitivos, compostos e complexos.

Conceito

A escolha do algoritmo para a solução de um determinado problema depende do problema, dos recursos disponíveis e da característica dos dados.

Nem sempre o algoritmo mais eficiente é a melhor escolha. Pode ser que o custo envolvido na implementação do algoritmo, e os recursos necessários para garantir que ele realmente seja o mais rápido não compense o benefício adquirido. Isto é mais perceptível quando a lista de itens que precisa ser ordenada é pequena ou quando esta lista tem um nível de entropia baixo, poucos itens estão fora da ordem. Ou ainda quando o conjunto de dados tem pouca variação. Podem ser muitos dados mas poucos valores diferentes.

Algoritmos diferentes terão eficiência diferente em dados diferentes. Isso quer dizer que um algoritmo pode ser muito eficiente em casos onde os dados estão arranjados em um conjunto com pouca entropia e não ter utilidade nenhuma em casos onde os dados estão na ordem inversa da desejada, ou seja em entropia máxima. A entropia, neste cenário se refere ao percentual de dados que está fora da ordem desejada. Dados que já se encontrem na ordem desejada terão entropia zero dados totalmente ordenados na ordem invertida, terão entropia 1, ou 100%.

Conceito - aplicação

Fora do mundo acadêmico é raro utilizarmos um algoritmo para ordenar uma lista de números, palavras ou frases.

O comum é que o dado que precisa ser ordenado esteja representado por uma estrutura composta de vários campos, *record* com um ou mais campos chaves, *keys*, a partir dos quais realizaremos a ordenação. Este é o caso típico que encontramos em bancos de dados sejam eles relacionais, documentais ou gráficos. Por exemplo, um tabela para dados de alunos pode conter campos como *nome*, *cpf* e *data de nascimento*. Cada um com suas próprias características. Um campo *nome*, por exemplo, permite o uso de um algoritmo de ordenação alfabética que não terá utilidade para o campo *data de nascimento*. Este cenário implica em

procedimentos especiais tanto para a alocação de memória e manipulação destes dados quanto para a criação dos algoritmos de ordenação.

Exemplo visual

Um dos benefícios da internet é poder ver animações de conceitos. Neste contexto Viktor Bohush publicou no YouTube o vídeo *Visualization and Comparison of Sorting Algorithms* que permite uma comparação visual de nove algoritmos de ordenação que pode ser visto na Figura 1.

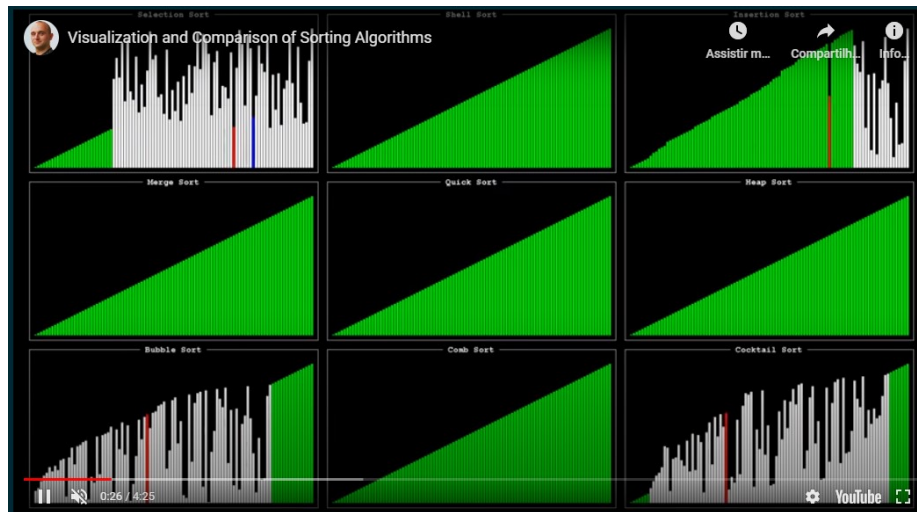


Figure 1: Fonte:(BOHUSH,2014) Animação de algoritmos de ordenação

Nesta visualização Bohoshu optou por testes diferentes: dados randômicamente organizados, onde, para determinarmos a entropia precisaríamos de acesso a estes dados (*random*); dados cujo conjunto é formado por repetições do mesmo valor, com pouca diferenciação (*few unique*); dados com entropia 1 (*reversed*), que estão na ordem reversa e, finalmente, dados que estão quase alinhados, entropia muito baixa (*almost sorted*). Em alguns casos, o shell sort se destaca, em outros o merge sort ou o quick sort, até o insertion sort teve seu momento de glória.