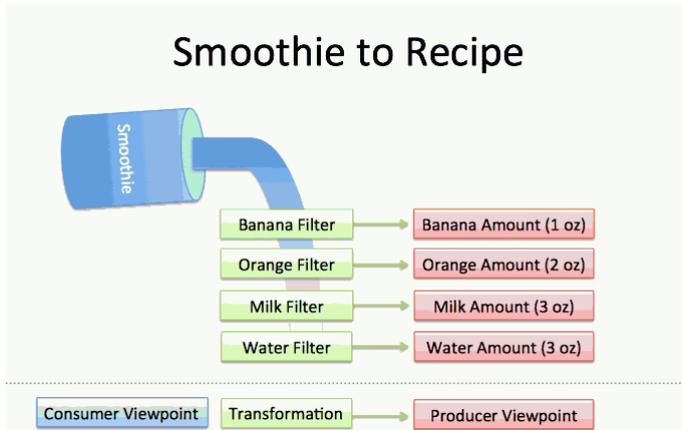


Esta folha está baseada no excelente livro BHARGAVA, Aditya. **Entendendo Algoritmos** - um guia ilustrado para programadores e outros curiosos. São Paulo: Novatec, 2017.

Transformada de Fourier

Fourier foi um cientista francês que acompanhou Napoleão na invasão do Egito e depois ficou por lá. É um grande matemático e físico. Trabalhou muito no conceito de calor e é um dos suportes da Revolução Industrial. A transformada de Fourier é um dos raros algoritmos que conseguem ser brilhantes, elegantes e ter milhares de formas de utilização (BHA17). A melhor analogia para o algoritmo vem do Better Explained: dada uma vitamina de frutas, a transformada de Fourier consegue transformá-lo em mamão, maçã, açúcar, sorvete etc. Ele é usado no algoritmo MP3 para transformar uma música em suas frequências individuais (e daí o MP3 exclui as frequências que são irrelevantes, por inaudíveis, por isso ele consegue comprimir o som). O compressor de imagens JPG faz a mesma coisa, mas com cores e regiões. Idem para o aplicativo Shazam que descobre qual é a música que está tocando, já que ele compara as frequências fundamentais da música e não a música em si. Veja a explicação da vitamina retirada do site

<https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/>



Algoritmos Paralelos

Como já se viu, o melhor algoritmo de ordenação tem custo $O(n \times \log_2 n)$. O que não foi dito naquela ocasião, mas que é verdade, é que este custo é para ambientes mono-processados, ou seja naqueles nos quais as operações são seriais, só é possível fazer uma operação de cada vez.

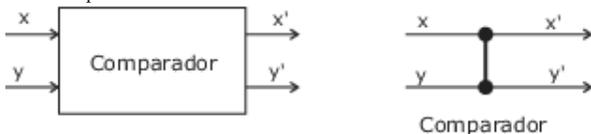
Suponha agora que você dispõe de um sistema que permita processamento paralelo (diversas operações a cada ciclo). Qual seria um bom algoritmo para obter uma ordenação mais rápida?

Trata-se do conceito de redes de ordenação. Usando-as, pode-se criar um algoritmo de ordenação de n elementos com custo $O(\log_2^2 n)$.

Para relembrar o significado dessas fórmulas, preencha:

Uma rede de comparações é um conjunto de fios e de comparadores. Um comparador é um dispositivo com duas entradas (x, y) e duas saídas (x', y') que executa as seguintes funções: $x' = \min(x, y)$ e $y' = \max(x, y)$

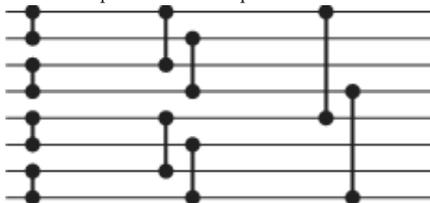
Um comparador pode ser apresentado como uma caixa (à moda LISP), ou mais praticamente como uma única linha vertical.



No desenho acima, as entradas aparecem à esquerda do comparador e as saídas à direita. Cada comparador opera com tempo $O(1)$, ou seja o tempo gasto entre a chegada das entradas e a obtenção das saídas é uma constante.

Uma rede de ordenação é um conjunto de comparadores interconectados por fios, de tal maneira que a saída de um comparador pode ser a entrada de outro.

Acompanhe no exemplo:



A rede de comparações acima, consegue responder qual é o maior e o menor elementos de uma entrada contendo 8 valores, usando apenas 3 comparações. Lembrando, 3 é o $\log_2 8$. Faça um chinês com os valores 2, 7, 1, 9, 3, 6, 4 e 8.

Existe um algoritmo paralelo que está se tornando muito popular, principalmente com a popularização de CPUs multi-núcleo. Trata-se do algoritmo

map-reduce (disponível no framework livre Apache Hadoop). É composto de duas partes: a função map e a função reduce. A primeira (map) pega uma função qualquer e um array e aplica esta função a cada elemento do array. A segunda, pega um array e transforma-o em um único número aplicando aos elementos do array uma dada função. Quando você tiver um conjunto de dados com bilhões de linhas o MapReduce pode fornecer uma resposta em minutos enquanto um BD tradicional pode levar horas.

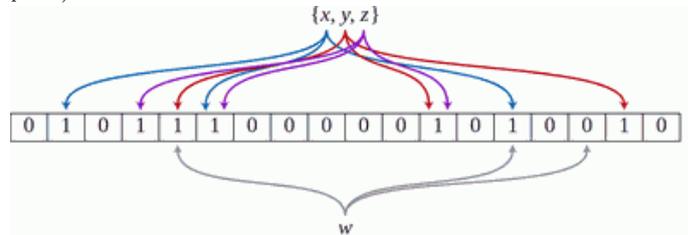
Filtros de Bloom

Uma estrutura de dados probabilística, inventada por Burton Bloom em 1970, que é usada para verificar se um elemento é membro de conjunto de maneira muito rápida e eficaz. O falso positivo (achar que o elemento é membro, mas ele não é) é possível – embora com probabilidade tão baixa quanto se queira, mas o falso negativo (achar que o elemento não é membro e ele é) nunca ocorre. Sua construção usa os seguintes artefatos:

- Um conjunto x_1, x_2, \dots, x_n de elementos de um universo X .
- Um vetor S de m bits originalmente todos valendo 0.
- Um conjunto de k funções hash, todas diferentes h_1, h_2, \dots, h_k para as quais dado um valor X devolvem um valor no domínio $(1, \dots, m)$ ou seja uma posição no vetor S .

Para agregar um elemento novo de X aplica-se a ele todas as funções k e o resultado dessas funções é tornado 1 em S . Para buscar se um novo elemento pertence ou não a X , a busca retorna "SIM" se todas as funções consultadas possuem um 1 em S .

Da maneira como o filtro é construído, é impossível eliminar um elemento. A solução pode ser um segundo filtro de Bloom contendo os eliminados. Veja no exemplo a seguir o funcionamento de um filtro de Bloom (tirado da wikipedia)



Aqui está o conjunto $\{x, y, z\}$. As flechas mostram as posições no vetor de bits dos resultados após a aplicação das funções de hash a cada um dos elementos. O candidato w não está no conjunto $\{x, y, z\}$ porque ele tem um ou mais valores de hash valendo 0.

Para você fazer

Conduza uma pesquisa na Internet e faça um resumo pessoal dos seguintes algoritmos/sites:

1. Troca de Chaves Diffie-Hellman
2. Algoritmo HyperLogLog
3. Algoritmo Flajolet-Martin
4. Algoritmo Hash e SimHash (um hash que tem sensibilidade local)
5. o site better explained
6. o algoritmo de similaridade do cosseno
7. o algoritmo da maior subsequência comum
8. algoritmo da mochila (aplicação do alg. guloso)

