

Bits & Bytes

Fabio Razzo
Galuppo



Uma breve introdução sobre algoritmo e eficiência

Começaremos por um pequeno exercício. Imagine que você recebeu uma tarefa para escrever um programa genérico que determine o lucro máximo num dado período de tempo, por exemplo o período de 30 dias indicado na Figura 1. Duas regras básicas: 1) você não pode comprar e vender no mesmo dia; 2) antes de vender é necessário comprar.



Figura 1: Preço da ação em função do tempo

Para calcular o lucro, dois vetores são disponibilizados, um contendo os maiores valores e outro com os menores, H e L respectivamente. Ambos indexados pelos dias. A função objetivo do problema recebe como parâmetro estes vetores e deve retornar o lucro máximo. Matematicamente, isto é um problema de maximização.

Uma vez que temos a função objetivo e suas restrições, elaborar um modelo, ou melhor um código em alguma linguagem de programação, para resolver o problema proposto, não parece ser complexo.

```
int l = 0, h = 0;
for (int i = 0; i < L.Length; ++i)
{
    for (int j = i + 1; j < H.Length; ++j)
    {
        if (H[j] - L[i] > H[h] - L[l])
        {
            h = j;
            l = i;
        }
    }
}
```

Figura 2: Algoritmo básico para resolução do problema proposto

Na Figura 2, é apresentado o trecho principal do algoritmo para resolver o problema de uma maneira simples (ou naive). Ou seja, são visitados todos os possíveis candidatos da solução para verificar qual satisfaz o problema proposto. Isto é denominado busca exaustiva ou por força bruta. A solução é implementada rapidamente, o que se traduz em produtividade. De fato, resolve o problema. No entanto, uma solução por força bruta não está livre de um fenômeno indesejado chamado explosão combinatória. Para entender este fenômeno, imagine que para 30 dias será necessário verificar 435 combinações, uma quantidade aproximadamente quadrática ($\sim 900/2$), ou de forma mais precisa, assintoticamente quadrática em relação a entrada (ou tamanho) do problema. A quantidade de combinações representa o esforço computacional despendido nesta maximização. Então, existe uma maneira eficiente para resolver este problema? Neste caso, a resposta é sim.

Existe uma solução que necessita visitar os vetores 30 vezes aproximadamente, esta é uma solução eficiente e linear, onde o esforço é proporcional ao tamanho do problema, note a relação 30 dias – 30 vezes. Você conseguiria escrever (ou imaginar) um algoritmo que solucione o problema proposto de forma linear? Certamente, ele é mais desafiador do que aquele encontrado por força bruta apresentado anteriormente. Se quiser uma possível resposta para este problema, eu disponibilizei minha solução linear no endereço <http://bit.ly/2fExaTO>, mas eu encorajo você encontrá-la.

Basicamente, nós gastaremos mais tempo para encontrar uma solução definitiva. Aquela velha frase ecoa: “O bom é inimigo do ótimo”. No entanto, trocamos uma complexidade de 435 por 30, o que representa uma proporção de 14,5 vezes – olha que investimento! Isto se traduz em economia de processamento, e consequentemente de gasto, de energia, ciclos de CPU, entre outros. Resumindo, em economia de dinheiro. Considerando que a quantidade gasta da energia elétrica é paga em função da energia de processamento – se processa mais, gasta mais. Ou, se considerar um mainframe, onde cobrança pode ser feita por ciclos de processamento. Ou, se considerar um blockchain público, onde a cobrança em moedas virtuais é feita pela quantidade de processamento de um smart contract. Exemplos não faltarão. Assim como na Matemática aplicada, a Computação também possui uma forte ligação com os problemas de otimização (maximização ou minimização) e suas complexidades.

Vamos concluir com uma reflexão sobre eficiência: se prezarmos apenas por uma solução rápida ou produtiva, estaremos, de certa forma, “perdendo” dinheiro indiretamente por causa da ineficiência inerente de certas soluções computacionais. Programar de forma simplista parece ser fácil, porém programar eficientemente é uma nobre disciplina, ela requer criatividade, conhecimento e dedicação. E pode produzir “ganhos” indiretamente excepcionais.