

Análise de caso médio do Quicksort

Abaixo nós temos o pseudo-código do algoritmo Quicksort

```

func Quicksort (L, inicio, fim)

    Se ( inicio ≥ fim)

        Retorna

    k ← Partição (L, inicio, fim)

    Quicksort (L, inicio, k)

    Quicksort (L, k+1, fim)

```

Só que dessa vez, não é possível anotá-lo.

Porque a gente não sabe o tamanho das partes produzidas pela função Partição().

Quer dizer, no pior caso uma das partes fica vazia, e a outra tem $n - 1$ elementos.

Então, a gente pode anotar o código assim

```

T(n) =   func Quicksort (L, inicio, fim)

O(1)  |   Se ( inicio ≥ fim)
      |       Retorna
O(n)  |   k ← Partição (L, inicio, fim)
T(n-1) ← Quicksort (L, inicio, k)
T(0)  ← Quicksort (L, k+1, fim)

```

E nós obtemos a seguinte equação de recorrência

$$T(n) = T(n - 1) + O(n)$$

Agora, é só desenrolar o rocambole

$$\begin{aligned} T(n) &= T(n - 1) + n \\ &= T(n - 2) + (n - 1) + n \\ &= T(n - 3) + (n - 2) + (n - 1) + n \\ &= \dots \\ &= T(1) + 2 + 3 + \dots + (n - 1) + n \\ &= 1 + 2 + 3 + \dots + (n - 1) + n \\ &= O(n^2) \end{aligned}$$

O problema é que a análise de pior caso não é muito informativa nesse caso.

Porque na prática o pior caso não acontece.

Então o que nós precisamos é de uma análise de caso médio.

Mas, o que é o caso médio?

Bom, ele é a média aritmética de todos os casos.

E quais são todos os casos?

Bom, existe o pior caso



E existe o segundo pior caso



E existe o terceiro pior caso



Quer dizer, são dois casos de cada tipo.

E em todos eles, nós temos duas chamadas recursivas.

Então, quando a gente soma tudo dá

$$2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2) + 2 \cdot T(n - 1)$$

E a média aritmética fica assim

$$\frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2) + 2 \cdot T(n - 1)}{n}$$

Agora, a gente pode escrever a equação de recorrência assim

$$T(n) = \frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2) + 2 \cdot T(n - 1)}{n} + n$$

Só que isso não é um rocambole que a gente consegue desenrolar.

Então agora é a hora para uma esperteza.

Quer dizer, a esperteza é tentar ver o caso $n - 1$ dentro do caso n

E para isso, a gente foca a atenção nessa parte aqui

$$T(n) = \underbrace{\frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2) + 2 \cdot T(n - 1)}{n}}_{*} + n$$

Daí, o primeiro passo é separar o último termo

$$T(n) = \frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2)}{n} + \frac{2 \cdot T(n - 1)}{n} + n$$

Depois, multiplicar e dividir por $n - 1$

$$T(n) = \frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2)}{n} \cdot \frac{n - 1}{n - 1} + \frac{2 \cdot T(n - 1)}{n} + n$$

E depois trocar os denominadores, para ver o seguinte

$$T(n) = \underbrace{\frac{2 \cdot T(0) + 2 \cdot T(1) + \dots + 2 \cdot T(n - 2)}{n - 1} \cdot \frac{n - 1}{n}}_{T(n - 1) - (n - 1)} + \frac{2 \cdot T(n - 1)}{n} + n$$

Agora, a gente pode reescrever a recorrência assim

$$T(n) = \left[T(n - 1) - (n - 1) \right] \cdot \frac{n - 1}{n} + \frac{2 \cdot T(n - 1)}{n} + n$$

depois reescrever assim

$$T(n) = T(n - 1) \cdot \frac{n + 1}{n} - \frac{(n - 1)^2}{n} + n$$

e depois assim

$$T(n) = T(n - 1) \cdot \frac{n + 1}{n} + 2 - \frac{1}{n}$$

e depois assim

$$T(n) = T(n - 1) \cdot \frac{n + 1}{n} + 2$$

— (*jogando fora um termo que não vai fazer diferença, e reduzindo a constante ...*)

Agora a gente tem um rocambole que dá para desenrolar.

Então a gente vai fazer isso

$$\begin{aligned} T(n) &= T(n - 1) \cdot \frac{n + 1}{n} + 1 \\ &= \left[T(n - 2) \cdot \frac{n}{n - 1} + 1 \right] \cdot \frac{n + 1}{n} + 1 \\ &= T(n - 2) \cdot \frac{n + 1}{n - 1} + \frac{n + 1}{n} + 1 \\ &= T(n - 3) \cdot \frac{n + 1}{n - 2} + \frac{n + 1}{n - 1} + \frac{n + 1}{n} + 1 \\ &= \dots \\ &= T(0) \cdot \frac{n + 1}{1} + \frac{n + 1}{2} + \dots + \frac{n + 1}{n - 1} + \frac{n + 1}{n} + 1 \end{aligned}$$

Daí, colocando o $n + 1$ em evidência a gente obtém o seguinte

$$T(n) = 1 + (n + 1) \cdot \left[1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n - 1} + \frac{1}{n} \right]$$

Aqui, a gente reconhece a soma harmônica dentro dos colchetes.

E todo mundo sabe que essa soma vale (aprox.) $\log n$ — (*ou será que não?*)

Então agora, a gente pode concluir que

$$T(n) = O(n \log n)$$

Legal, não é?