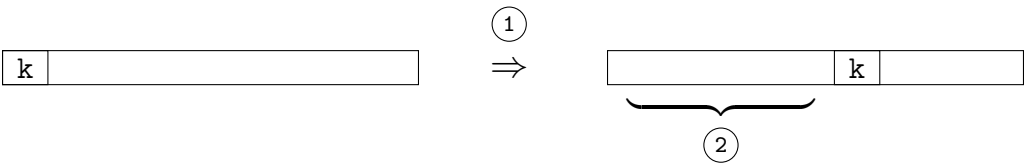


O algoritmo quickMediana

Existe uma ideia simples que funciona bem na prática.

A ideia consiste em utilizar a mesma estratégia do algoritmo quicksort

- 1. primeiro, a gente particiona a lista
- 2. e depois, a gente continua procurando no lado certo da partição



```
Algoritmo quickMediana ( L, inicio, fim )
{
    k <-- Particao ( L, inicio, fim )

    Se ( k = n/2 + 1 )    M <-- L[k]

    Se ( k > n/2 + 1 )    M <-- quickMediana (L, inicio, k-1)

    Se ( k < n/2 + 1 )    M <-- quickMediana (L, k+1, fim)

    Retorna ( M )
}
```

No pior caso, esse algoritmo é uma porcaria — (*porque?*)

Mas, ele executa em tempo médio $O(n)$.

Porque?

Bom, a ideia é utilizar a função **Partição-Boa** ().

Essa função executa em tempo médio $O(n)$.

E garante que o maior lado da partição tem tamanho no máximo $3n/4$.

Daí, a anotação do código fica assim

```
Algoritmo quickMediana ( L, inicio, fim )
{
    k <-- Particao-Boa ( L, inicio, fim )

    Se ( k = n/2 + 1 )    M <-- L[k]

    Se ( k > n/2 + 1 )    M <-- quickMediana (L, inicio, k-1)

    Se ( k < n/2 + 1 )    M <-- quickMediana (L, k+1, fim)

    Retorna ( M )
}
```

$O(n)$ ←
 $O(1)$ |
 $T(3n/4)$ ←
 $T(3n/4)$ ←
 $O(1)$ |

Daí, como apenas uma chamada recursiva é realizada, nós obtemos a seguinte recorrência

$T(n) = T\left(\frac{3n}{4}\right) + n$

Para obter a solução, nós desenrolamos a recorrência

$$\begin{aligned} T(n) &= T\left(\frac{3n}{4}\right) + n \\ &= T\left(\frac{3^2n}{4^2}\right) + \frac{3n}{4} + n \\ &= T\left(\frac{3^3n}{4^3}\right) + \frac{3^2n}{4^2} + \frac{3n}{4} + n \\ &= \dots \\ &< n \cdot \left[1 + \frac{3}{4} + \frac{3^2}{4^2} + \frac{3^3}{4^3} + \dots \right] \end{aligned}$$

Quer dizer, a gente chegou em uma soma de PG outra vez.

E a gente resolve isso assim

$$\begin{aligned} S &= 1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \left(\frac{3}{4}\right)^3 + \dots \\ &= 1 + \frac{3}{4} \cdot \underbrace{\left[1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \dots \right]}_S \end{aligned}$$

donde

$$S = 1 + \frac{3}{4} \cdot S$$

donde

$$S = 4$$

Daí que, o nosso algoritmo executa em tempo médio

$$T(n) = 4n = O(n)$$

Mas, no pior caso ele é uma porcaria.

Logo, ainda deve ser possível conseguir algo melhor do que isso.