Quicksort e quicksort aleatorizado

QS clássico

```
Quick sort clássico
-------------------
Dados: um vector x[a..b] a ordenar

1    Se o número de elementos do vector (b-a+1) é 0 ou 1:
       HALT
2    Escolhe um "pivot", isto é um dos x[p], com a <= p <= b
     Seja piv=x[p]
3    (Split) particiona-se o vector x[a..b] em 3 partes:
       x[a..m-1]: com elementos x[i]<piv
       x[m]:      com o valor piv, x[m]=piv
       x[m+1..b]: com elementos x[i]>piv
4    Recursivamente aplica-se o algoritmo a x[a..m-1]
5    Recursivamente aplica-se o algoritmo a x[m+1..b]
```

Análise da eficiência

- Hipóteses sobre os dados: distintos, todas as $n!$ permutações com a mesma probabilidade
- Análise da eficiência no pior caso
- Análise da eficiência no pior médio:

Probabilidade da ordem do pivot. Partes em que se divide `v[a..b]` e sua probabilidade. Recorrência do valor esperado do número de comparações.

$$\begin{cases} t(0) = 0 \\ t(1) = 0 \\ t(n) = (n-1) + \frac{1}{n}\sum_{i=0}^{n-1}(t(i) + t(n-i-1)) \end{cases}$$

Note-se que $t(0) = 0$ e que cada um dos restantes $t(i)$ ocorre duplicado.

$$\begin{aligned} t(4) &= 3 + [(t(0) + t(4)) + (t(1) + t(3)) + (t(2) + t(2)) + (t(3) + t(1)) + (t(4) + t(0))] \\ &= 3 + 2[(t(1) + t(2) + t(3) + t(4))] \end{aligned}$$

Assim a equação geral da recorrência pode escrever-se

$$t(n) = (n-1) + \frac{2}{n}\sum_{i=1}^{n-1} t(i)$$

Suspeita. $c(n) \leq n \ln n$ para c apropriado. Porque é suspeita?

Demonstração e determinação de c. Casos base $n = 0, 1$ triviais. Passo de indução.

$$
\begin{aligned}
t(n) &= (n-1) + \frac{2}{n}\sum_{i=1}^{n-1} t(i) && (1)\\
&\leq (n-1) + \frac{2c}{n}\sum_{i=1}^{n-1} i\ln i && (2)\\
&\leq (n-1) + \frac{2c}{n}\int_1^n x\ln x dx && (3)\\
&\leq (n-1) + \frac{2c}{n}\left(\frac{1}{2}n^2\ln n - \frac{n^2}{4} + \frac{1}{4}\right) && (4)\\
&\leq cn\ln n && (5)
\end{aligned}
$$

desde que $c \geq 2$.
Justificação

QS aleatorizado

```
Quick sort aleatorizado
-----------------------
Dados: um vector x[a..b] a ordenar

1    Se o número de elementos do vector (b-a+1) é 0 ou 1:
       HALT
2    É escolhido aleatoriamente e uniformemente (com igual
     probabilidade) um índice p de {a,a+1,...,b}.
     Seja piv=x[p]
3    Particiona-se o vector x[a..b] em 3 partes:
       x[a..m-1]: com elementos x[i]<piv
       x[m]:      com o valor piv, x[m]=piv
       x[m+1..b]: com elementos x[i]>piv
4    Recursivamente aplica-se o algoritmo a x[a..m-1]
5    Recursivamente aplica-se o algoritmo a x[m+1..b]
```

Análise da eficiência e conclusões

- A alteração.
- Qualquer que seja x, a probabilidade de o pivot ser o menor, o segundo menor... é $1/n$.
- Conclusão. Qualquer que seja a distribuição dos x o comportamento éidêntico ao clássico no caso médio.