

O algoritmo quicksort

Imagine que você tem uma lista desordenada.

Por exemplo,

V

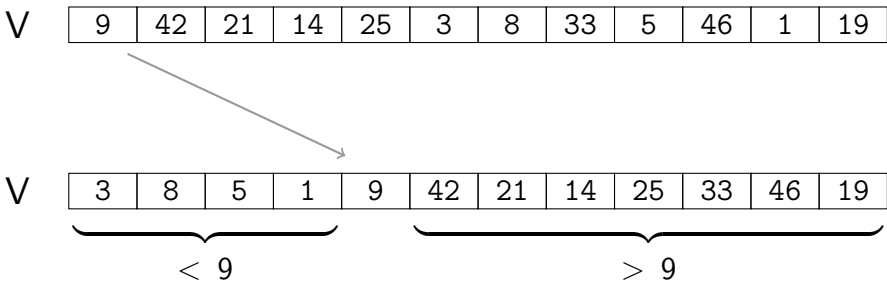
9	42	21	14	25	3	8	33	5	46	1	19
---	----	----	----	----	---	---	----	---	----	---	----

Como é que a gente coloca essa lista em ordem?

Bom, um dos algoritmos mais rápidos para fazer isso é o *Quicksort*.

E a sua ideia é bem simples.

Quer dizer, ele começa aplicando o procedimento de partição, utilizando o primeiro elemento como pivô



E agora, é preciso ordenar duas faixas da lista: L[1..4] e L[6..12].

A ideia, claro, é fazer a mesma coisa outra vez.

E nós vamos utilizar uma pilha para organizar essa computação.

Quer dizer,

1. No início, a gente coloca na pilha a tarefa de ordenar a lista inteira

2. Daí, a cada passo, a gente
 - remove uma tarefa da pilha
 - realiza a partição
 - e coloca as duas tarefas resultantes na pilha

— (a menos que a tarefa seja trivial)

3. E continua fazendo isso até que a pilha esteja vazia

A coisa fica assim

```
func Quicksort ( L )
  Push ( (1..N), P )           // Coloca a lista inteira na pilha
  Enquanto ( P não está vazia )
    (i..f) = Pop ( P )
    Se ( i < f )                // Se existe mais de um elemento na faixa
      k = Partição ( L, i, f ) // Particiona a faixa
      Push ( (i..k-1), P )      //
      Push ( (k+1..f), P )      // e empilha as duas faixas resultantes
```