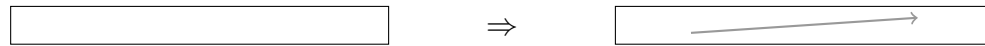


A ordenação por seleção é uma porcária

Hoje nós vamos olhar para o problema da ordenação de uma lista um pouco mais de perto



Quer dizer, a ideia natural para fazer isso é:

- localizar o maior elemento, e colocar na última posição
- localizar o segundo maior elemento, e colocar na penúltima posição
- e por aí vai ...

```
func Ord-Selecao ( L )  
  Para i = N Até 2  
    M = L[1];    posM = 1  
    Para j = 2 Até i  
      Se ( L[j] > M )  
        M = L[j];    posM = j  
    Troca ( L, i, j )
```

Mas, esse algoritmo é uma porcária.

Porque?

Bom, porque ele executa em tempo $O(n^2)$.

Porque?

Bom, porque ele percorre a lista n vezes.

Sim, mas porque isso é uma porcária?

Bom, porque isso acaba levando a gente a fazer a mesma coisa várias vezes.

Por exemplo, imagine que a lista começa assim

V

17	30								
----	----	--	--	--	--	--	--	--	--

onde o 17 e o 30 não são os maiores elementos da lista.

Então, a cada vez que o algoritmo percorrer a lista, ele vai comparar esses dois elementos para descobrir que o 30 é maior que o 17.

E se a lista tiver tamanho 1.000.000, o algoritmo vai fazer essa comparação 1.000.000 de vezes.

Isso não faz sentido, não é?

Hoje nós vamos resolver isso — (*três vezes ...*)