

Aula de 28 de fevereiro

Algoritmos elementares de ordenação

Por que elementares?

- utilizam técnicas bem simples de programação
 - laços
 - condições para comparação
 - são eficazes, isto é, ordenam
 - mas...
 - não são eficientes para grandes volumes de dados

São 3 os algoritmos elementares de ordenação

- bubblesort
- insertionsort
- selectionsort

1 = veg

| | | | | |
|---|---|---|---|---|
| 3 | 4 | 1 | 5 | 2 |
| 0 | 1 | 2 | 3 | 4 |

| | | | | |
|---|---|---|---|---|
| 3 | 1 | 4 | 5 | |
| 0 | 1 | 2 | 3 | 4 |

| | | | | |
|---|---|---|---|---|
| 3 | 1 | 4 | 2 | 5 |
| 0 | 1 | 2 | 3 | 4 |

veg = 1
 0 - 1
 1 - 2
 2 - 3
 3 - 4

2 = veg

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 4 | | |
| 0 | 1 | 2 | 3 | 4 |

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 2 | 4 | 5 |
| 0 | 1 | 2 | 3 | 4 |

| | | | | |
|---|---|---|---|---|
| 1 | 3 | | | |
| 0 | 1 | 2 | 3 | 4 |

veg = 2
 0 - 1
 1 - 2
 2 - 3

3 =
very



| | | | | |
|---|---|---|---|---|
| 1 | 3 | 2 | 4 | 5 |
| 0 | 1 | 2 | 3 | 4 |

0-1
1-2

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 0 | 1 | 2 | 3 | 4 |

4 = very



| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 0 | 1 | 2 | 3 | 4 |

0-1

Algoritmo do Bubblesort

- Considerando um vetor de tamanho n
- laço de 1 a $n-1$
 - laço para posições iniciando em 0 e finalizando em ????

```
for (i = 1; i < n; i++)  
    for (j = 0; j < n-i-1; j++)  
        if (v[j] > v[j+1])  
            troca(v[j], v[j+1])
```

$n = 5$

$i = 1$

0-1
1-2
2-3
3-4

$i = 2$

0-1
1-2
2-3

$i = 3$

0-1
1-2

$i = 4$

0-1

| | | | |
|---|---------------|---|---------------|
| 0 | $\frac{4}{0}$ | 4 | $\frac{4}{1}$ |
| 1 | $\frac{4}{0}$ | 5 | $\frac{4}{1}$ |
| 2 | $\frac{4}{0}$ | 6 | $\frac{4}{1}$ |
| 3 | $\frac{4}{0}$ | 7 | $\frac{4}{1}$ |

$$n \% 4 \in \{0, 1, 2, 3\}$$

```
v = (int *) malloc (n * sizeof(int));
```

casting
conversão de
tipos

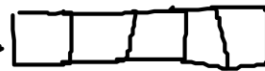
↓
quantas
caixinhas

↳ tamanho de cada
caixinha

memory allocation

região
variáveis estáticas

região de variáveis dinâmicas



v →

n 5