Ordenação - Bubblesort

Estrutura de Dados e Algoritmos – Ciência da Computação



Prof. Daniel Saad Nogueira Nunes

IFB – Instituto Federal de Brasília, Campus Taguatinga Bubblesort Análise



Sumário

Bubblesort



Bubblesort

Bubblesort

- O Bubblesort, em cada iteração, lê o vetor da esquerda para a direita e troca os elementos se v[i]>v[i+1].
- Como consequência disso, os maiores elementos são colocados em sua posição devida após cada iteração.
- ullet Observe que são necessárias n-1 iterações para o algoritmo ordenar a sequência original, sendo que cada iteração precisa passar por toda a sequência.
- É possível implementar uma otimização que interrompe as varreduras do vetor assim que é detectado que o vetor já está ordenado, mas isso não impacta no pior caso.

Bubblesort Análist



Bubblesort

11	17	23	2	7	29	3	13	5	19
11	17	2	23	7	29	3	13	5	19
11	17	2	7	23	29	3	13	5	19
11	17	2	7	23	3	29	13	5	19
11	17	2	7	23	3	13	29	5	19
11	17	2	7	23	3	13	5	29	19
44	17	2	7	23	2	12	_	40	20
П	17		/	23	J	13	Э	פו	29
11	17		1	23	3	13	5	19	29
	2		7						
11	2		7	23	3	13		19	29
11 11	2	17	7 17	23	3	13 13	5	19 19	29 29
11 11 11	2	17 7	7 17	23 23 3	3	13 13 13	5 5 5	19 19 19	29 29
11 11 11 11 11	2 2	17 7 7	7 17 17	23 23 3 3	3 3 23	13 13 13 23	5 5 5 5	19 19 19	29 29 29 29

Bubblesort Análist Análist



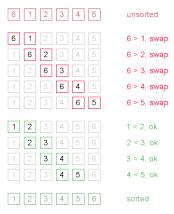
Bubblesort

		7							
2	11	7							
2	7	11	17	3	13	5	19	23	29
2									
2	7	11	3	13	17	5	19	23	29
2	7	11	3	13	5	17	19	23	29
2	7	3	11	13	5	17	19	23	29
2	7	3	11 11	13 5	5 13	17 17	19 19	23 23	29 29
2	7	3	11	5	13	17	19	23	29
2	7	7	11 11	5	13 13	17 17	19 19	23 23	29 29
2	7	7	11 11	5	13 13	17 17	19 19	23 23	29 29
2 2	7 3 3	7	11 11 5	5 11	13 13 13	17 17 17	19 19 19	23 23 23	29 29 29
2	7 3 3	3775	1157	5 11 11	13 13 13	17 17 17	19 19 19	23232323	29 29 29

Bubblesort Análise



Bubblesort



Bubblesort Análise



Bubblesort



Análise



Bubblesort

```
void swap(int *v, int i, int j) {
           int t = v[i];
           v[i] = v[j];
           v[i] = t;
 5
       }
7
       void bubble_sort(int *v, size_t size) {
           int swapped = 1;
           for (int i = 0; i < size - 1 && swapped; i++) {
10
               swapped = 0;
               for (int j = 0; j < size - i - 1; j++) {
11
                   if (v[i] > v[i + 1]) {
12
                       swap(v, j, j + 1);
13
14
                       swapped = 1;
15
16
               }
17
18
```

ubblesort Análise



Sumário

2 Análise



Bubblesort

Análise

No pior caso, são necessários n-1 iterações sobre a sequência original. Na iteração i são realizadas n-1-i comparações ao todo. Portanto, o custo do algoritmo é dado como:

$$\sum_{i=0}^{n-1} i = 1 + 2 + \ldots + n - 1 \in \Theta(n^2)$$

In-place	Estável
✓	√