



# Ordenação: ShellSort

---

**Prof. Túlio Toffolo**

<http://www.toffolo.com.br>

**BCC202 – Aula 16**

Algoritmos e Estruturas de Dados I

- Proposto por Shell em 1959.
- É uma extensão do InsertSort.
- Problema com o algoritmo de ordenação por inserção:
  - Troca itens adjacentes para determinar o ponto de inserção.
  - São efetuadas  $n - 1$  comparações e movimentações quando o menor item está na posição mais à direita no vetor.
- O método de Shell contorna este problema permitindo trocas de registros distantes um do outro.

# ShellSort

- Os itens separados de **h** posições são rearranjados.
- Todo h-ésimo item leva a uma seqüência ordenada.
- Tal seqüência é dita estar h-ordenada.

# ShellSort

- Exemplo de utilização:

	1	2	3	4	5	6
Chaves iniciais:	<i>O</i>	<i>R</i>	<i>D</i>	<i>E</i>	<i>N</i>	<i>A</i>
$h = 4$	<i>N</i>	<i>A</i>	<i>D</i>	<i>E</i>	<i>O</i>	<i>R</i>
$h = 2$	<i>D</i>	<i>A</i>	<i>N</i>	<i>E</i>	<i>O</i>	<i>R</i>
$h = 1$	<i>A</i>	<i>D</i>	<i>E</i>	<i>N</i>	<i>O</i>	<i>R</i>

- Quando  $h = 1$ , Shellsort é igual ao algoritmo de inserção.

- Como escolher o valor de  $h$ ?
  - Para  $s=1$ :
    - **$h(s) = 1$**
  - Para  $s > 1$ :
    - **$h(s) = 3h(s - 1) + 1$**
- A sequência para  **$h$**  corresponde a 1, 4, 13, 40, 121, 364, 1.093, 3.280, ...
- Knuth (1973, p. 95) mostrou experimentalmente que esta sequência é difícil de ser batida por mais de 20% em eficiência.

# ShellSort

- Como escolher o valor de  $h$ ?

$$h = \lfloor \log_3(2n + 1) \rfloor$$

# **SHELLSORT** **IMPLEMENTAÇÃO**

# ShellSort

```
void shellSort(TItem *v, int n) {  
    int i, j, h = 1;  
    TItem aux;  
  
    do { h = h * 3 + 1; } while (h < n);  
  
    do {  
        h /= 3; // h = ( h - 1 ) / 3  
        for(i = h ; i < n ; i++) {  
            aux = v[i]; j = i;  
            while (v[j - h].chave > aux.chave) {  
                v[j] = v[j - h]; j -= h;  
                if (j < h) break;  
            }  
            v[j] = aux;  
        }  
    } while (h != 1);  
}
```



SHELLSORT

# ANÁLISE DO ALGORITMO

- Análise
  - A razão da eficiência do algoritmo ainda não é conhecida.
  - Ninguém ainda foi capaz de analisar o algoritmo.
  - A sua análise contém alguns problemas matemáticos muito difíceis.
  - A começar pela própria sequência de incrementos.
  - O que se sabe é que cada incremento não deve ser múltiplo do anterior.

- Análise
  - Conjecturas referente ao número de comparações para a sequência de Knuth:
  - Conjectura 1:
    - $C(n) = O(n^{1,25})$
  - Conjectura 2:
    - $C(n) = O(n (\ln n)^2)$

**SHELLSORT**

**VANTAGENS/DESVANTAGENS**

- Vantagens:
  - Shellsort é uma ótima opção para arquivos de tamanho moderado.
  - Sua implementação é simples e requer uma quantidade de código pequena.
- Desvantagens:
  - O tempo de execução do algoritmo é sensível à ordem inicial do arquivo.
  - O método **não é estável**.



Perguntas?

SHELLSORT  
**EXERCÍCIO**

# Exercício

- Dada a sequência de números:

**3 4 9 2 5 1 8 0 9**

Ordene em ordem crescente utilizando o algoritmo **ShellSort**, apresentado a sequência dos números e explicando cada passo do algoritmo.

- Utilize  $h = 1, 4, 13, 40, 121, 364, 1.093, 3.280, \dots$