

# Shell Sort

Ciência da Computação  
Laboratório de Ordenação e Pesquisa  
Prof. M.Sc. Elias Gonçalves

# Criação

- Proposto por Donald Shell em 1959.
- É uma extensão do Insertion Sort.

# Ideia

- Permitir trocas de registros distantes um do outro e não somente nas adjacências.
- Os itens da lista separados de **h** posições são rearranjados, gerando sequências ordenadas.
  - O elemento na posição **j** é comparado e trocado com o elemento na posição **j-h**.
  - O vetor resultante é composto de **h** arquivos.
  - Cada sequência está **h-ordenada**.
  - O valor de **h** é reduzido a cada passo, até atingir o valor **1**.

# Dinâmica

→ Ordenar o vetor:

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→ O vetor tem tamanho 8 e o  $h$  inicial tem valor 1. Então ao aplicar a fórmula de Knuth obteremos  $h = 13$ , mas ao iniciar o primeiro passo o  $h$  será atualizado para o valor 4, pois faz-se a divisão do seu valor atual por 3 nesse e a cada novo passo.

```
while(h < tam){  
    h = 3*h+1;  
}
```

→ Se aplicarmos o método de divisão direta do  $h$  pela metade do tamanho do vetor, obteremos  $h = 4$ , pois o  $h$  inicial recebe o tamanho do vetor:

$$h = \text{tam}/2, \text{ ou seja, } h = 8/2 = 4$$

# Dinâmica

→ Ordenar o vetor:

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→ Para cada iteração, o elemento na posição  $j$  é comparado e trocado com o elemento na posição  $j-h$

→  $h = 4$ ,  $j = 4$ :

$5 < 23$ , sim. Troca

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→  $h = 4$ ,  $j = 5$ :

$12 < 17$ , sim. Troca

5	17	10	19	23	12	2	22
0	1	2	3	4	5	6	7

5	12	10	19	23	17	2	22
0	1	2	3	4	5	6	7

# Dinâmica

→ Ordenar o vetor:

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→ Para cada iteração, o elemento na posição **i** é comparado e trocado com o elemento na posição **j-h**

→  $h = 4$ ,  $j = 6$ :

$2 < 10$ , sim. Troca

5	12	10	19	23	17	2	22
0	1	2	3	4	5	6	7

→  $h = 4$ ,  $j = 7$ :

$22 < 19$ , não. Mantém

5	12	2	19	23	17	10	22
0	1	2	3	4	5	6	7

5	12	2	19	23	17	10	22
0	1	2	3	4	5	6	7

# Dinâmica

→ Ordenar o vetor:

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→ Agora, ao aplicar a fórmula de Knuth obteremos o valor de  **$h = 1$** , pois faz-se a divisão do seu valor atual (4) por 3.

→ Nesse passo se aplicarmos a divisão direta do tamanho do vetor por dois, obteríamos  **$h = 2$** ;

$$h = h/2, \text{ ou seja, } h = 4/2 = 2$$

# Dinâmica

→ Ordenar o vetor:

23	17	10	19	5	12	2	22
0	1	2	3	4	5	6	7

→ Para cada iteração, o elemento na posição  $i$  é comparado e trocado com o elemento na posição  $j-h$

→ **A partir de agora, com  $h = 1$  o algoritmo tem funcionamento igual ao insertion sort.**

→  $h = 1, j = 1$ :

12 < 5, não. Mantém

10 < 12, sim. Troca

...

5	12	10	19	23	17	2	22
0	1	2	3	4	5	6	7



# Código

Usando a Fórmula de Knuth para calcular o h

```
void shell_sort(int v[], int tam){
    int i, j, aux;
    int h = 1;
    while(h < tam){ h = 3*h+1; }
    while(h > 1){
        h /= 3;
        for(i = h; i<tam; i++) {
            aux = v[i];
            j = i-h;
            while(j>=0 && aux < v[j]) {
                v[j+h] = v[j];
                j -= h;
            }
            v[j + h] = aux;
        }
    }
}
```

# Código

Usando o tamanho do vetor e a divisão do h para calcular o h

```
int shellSort(int v[], int tam) {  
    for (int h = tam/2; h > 0; h/= 2){  
        for (int i = h; i < tam; i += 1){  
            int aux = v[i];  
            int j;  
            for (j = i; j >= h && v[j-h] > aux; j -= h)  
                v[j] = v[j - h];  
            v[j] = aux;  
        }  
    }  
    return 0;  
}
```

# Biblioteca Virtual

CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas.  
**Introdução a Estruturas de Dados com Técnicas de Programação em C** (Capítulo 11);

DROZDEK, Adam. **Estrutura de dados e algoritmos em c++**  
(Capítulo 9);

MARKENZON, Lilian; SZWARCFITER, Jorge Luiz. **Estruturas de Dados e seus Algoritmos** (Capítulos: 7, 11 e 12);

PINTO, Rafael Albuquerque. **Estrutura de Dados** (Páginas 155 a 177).