

Sistemas de Informação

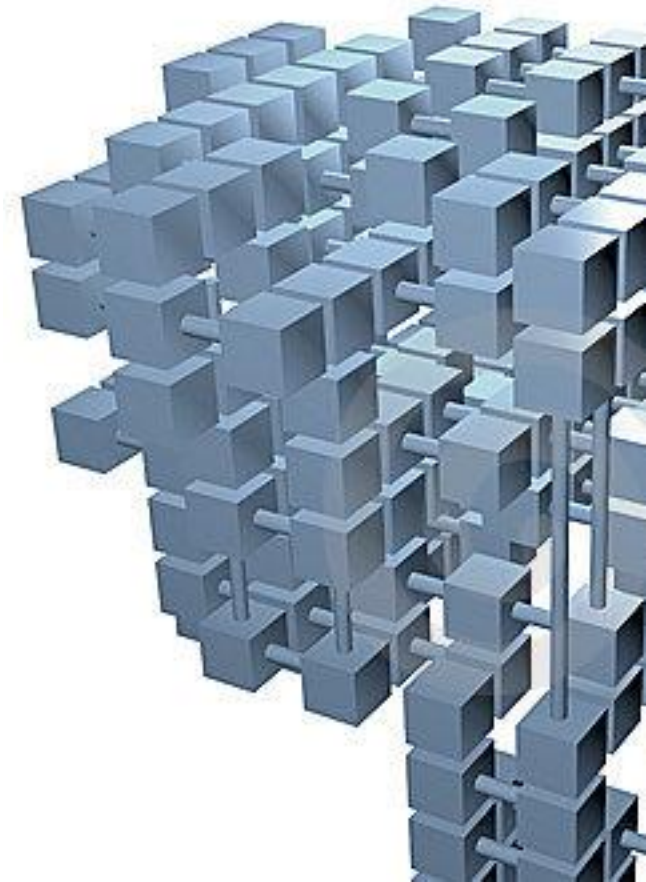
Estrutura de Dados

II

Ordenação por
Particionamento

Prof. Ivan José dos Reis Filho

ivanfilhoreis@gmail.com



Conteúdo da Aula

- Ordenação por Seleção
 - Ordenação por Inserção
 - Ordenação por Troca
- Já Vimos
- Ordenação por Particionamento
(Shell Sort)

Shell Sort

- Particiona as entradas na sequência:

$$3x+1:$$

$$1$$

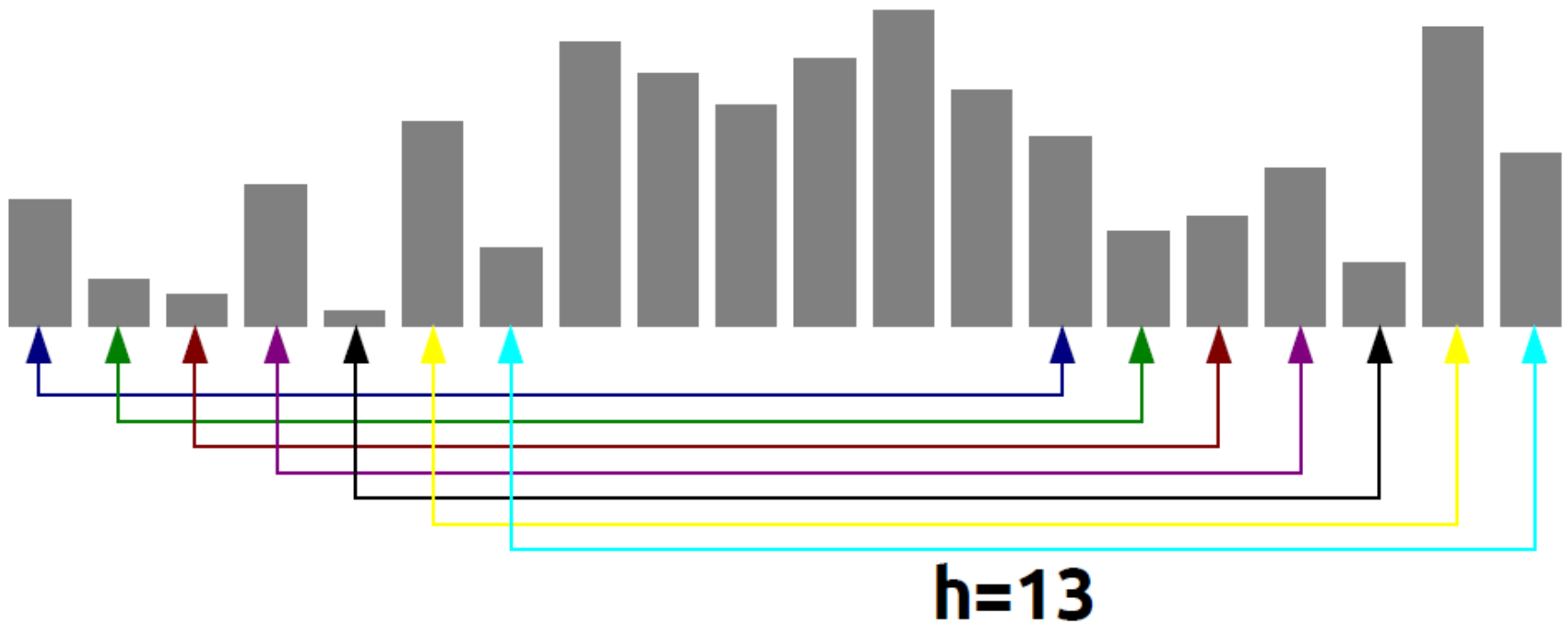
$$(3 \times 1) + 1 = 4$$

$$(3 \times 4) + 1 = 13$$

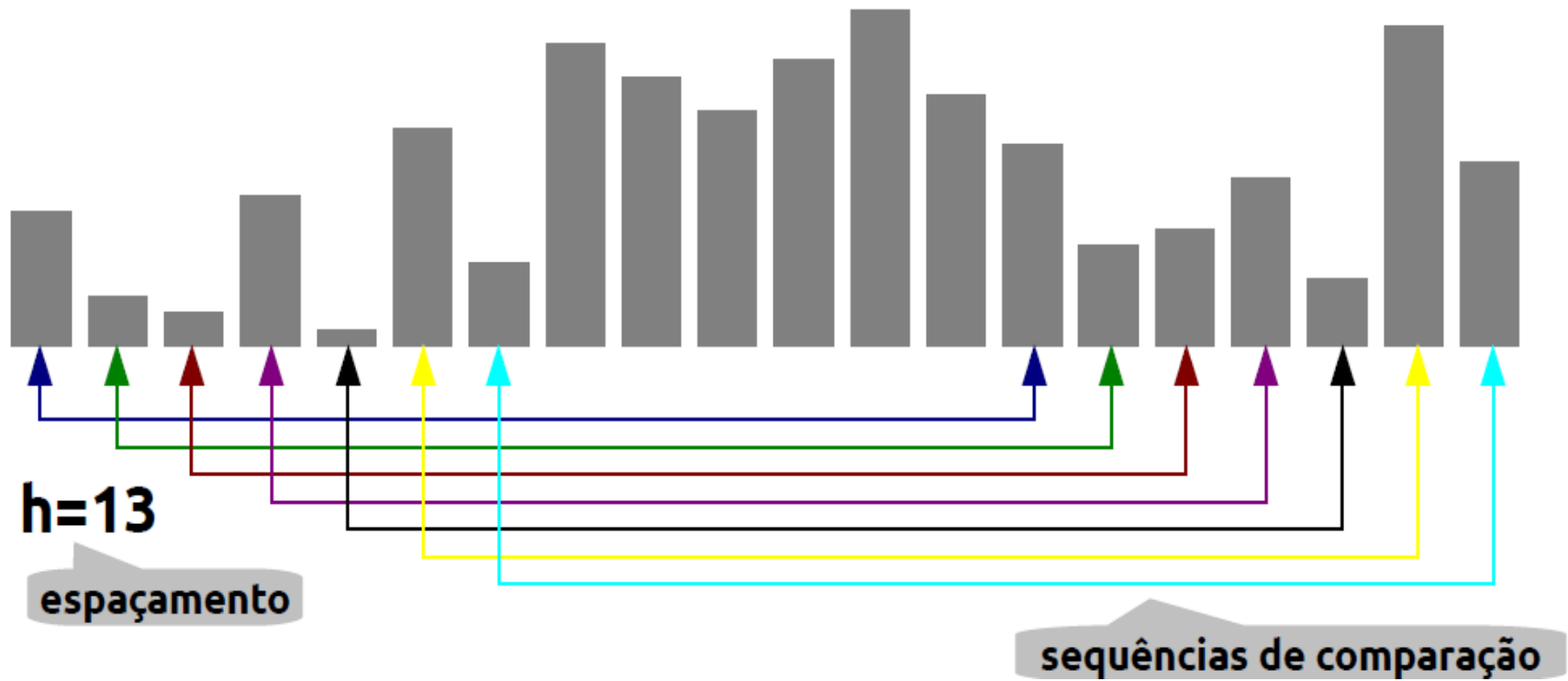
$$(3 \times 13) + 1 = 40$$

Shell Sort

- Percorrer a sequência e mover os elementos mais de uma posição por comparação (*h*sorting)
- Decrementar o valor de *h* e repetir o processo.

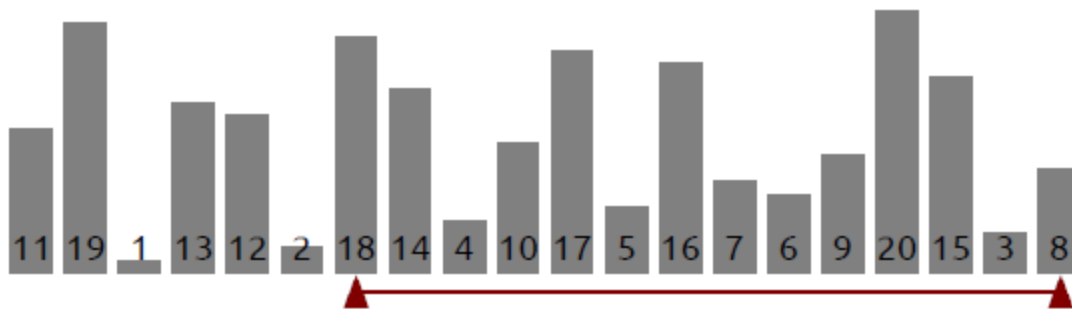


Shell Sort



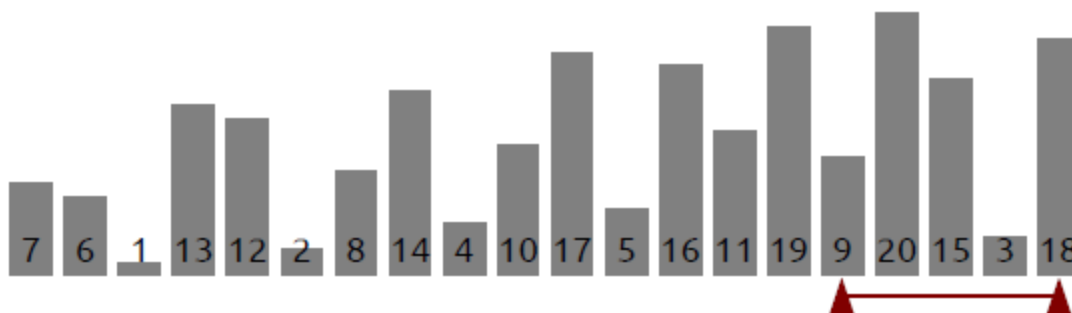
ShellSort

Entrada:



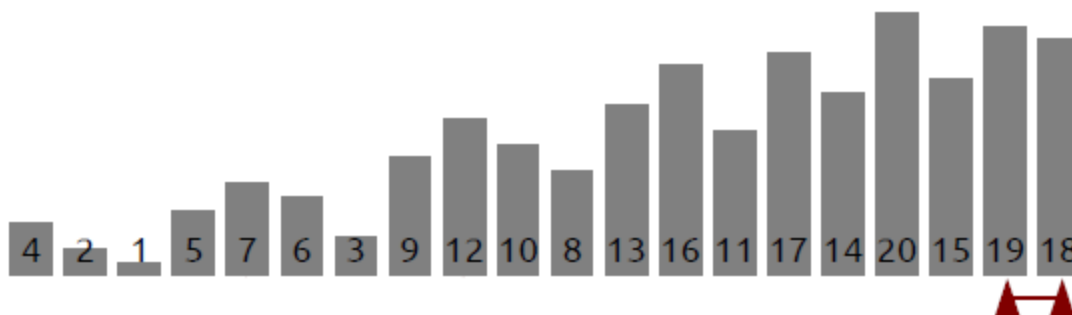
$h=13$

Passo 1:



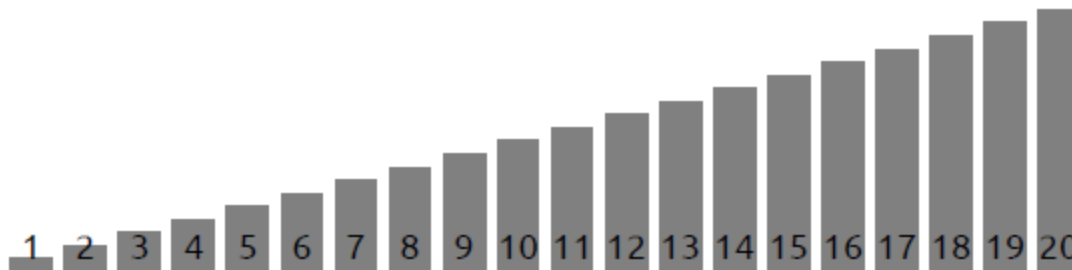
$h=4$

Passo 2:



$h=1$

Passo 3:



0-3

Shell Sort

```
template <class Item>
void shell(Item vetor[], int n)
{
    int h = 1;
    while( h < n/3 )
        h = 3*h + 1; // 1, 4, 13, 40, 121

    while (h >= 1)
    {
        for(int i = h; i < n; i++)
        {
            int j = i;
            while( j >= h && vetor[j-h] > vetor[j] )
            {
                swap( vetor[j-h], vetor[j] );
                j = j - h;
            }
        }
        h = h/3;
    }
}
```

i

controla a iteração dentro
de um espaçamento
(número de sequências de
comparação)

h

controla o espaçamento
das sequências de
comparações

j

controla a iteração dentro
de uma sequência de
comparação

```
template <class Item>
void swap(Item &A, Item &B)
{ Item t = A ; A = B; B = t; }
```

```

template <class Item>
void shell(Item vetor[], int n)
{
    int h = 1;
    while( h < n/3 )
        h = 3*h + 1; // 1, 4, 13, 40, 121

    while (h >= 1)
    {
        for(int i = h; i < n; i++)
        {
            int j = i;
            while( j >= h && vetor[j-h] > vetor[j] )
            {
                swap( vetor[j-h], vetor[j] );
                j = j - h;
            }
        }
        h = h/3;
    }
}

```

i

controla a iteração dentro
de um espaçamento
(número de sequências de
comparação)

h

controla o espaçamento
das sequências de
comparações

j

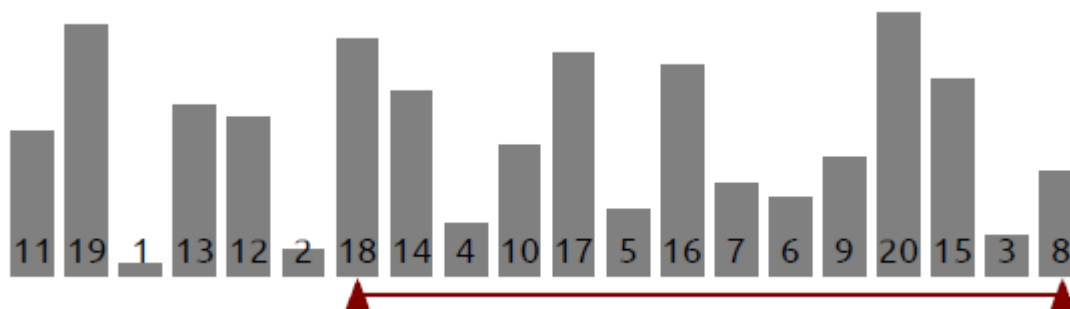
controla a iteração dentro
de uma sequência de
comparação

```

template <class Item>
void swap(Item &A, Item &B)
{ Item t = A ; A = B; B = t; }

```

Entrada:



ShellSort

h=13

Shell Sort

- Qual espaçamento usar?
- Quantas comparações são executadas?
- Quantas trocas são utilizadas?

```
template <class Item>
void shell(Item vetor[], int n)
{
    int h = 1;
    while( h < n/3 )
        h = 3*h + 1; // 1, 4, 13, 40, 121

    while (h >= 1)
    {
        for(int i = h; i < n; i++)
        {
            int j = i;
            while( j >= h && vetor[j-h] > vetor[j] )
            {
                swap( vetor[j-h], vetor[j] );
                j = j - h;
            }
        }
        h = h/3;
    }
}
```

DÚVIDAS?

