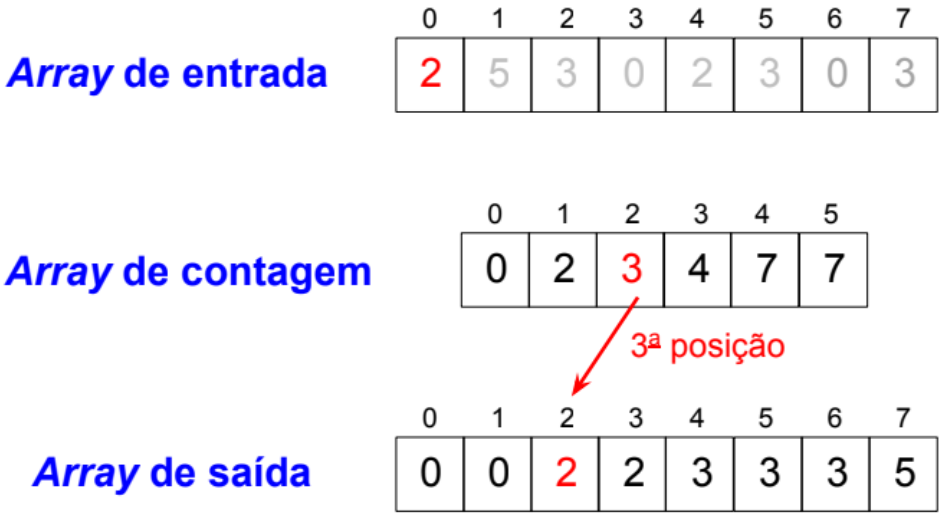
Trabalho Teórico 7

Unidade IV:

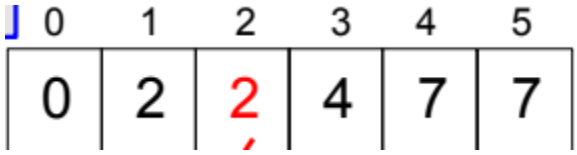
Ordenação Interna - Shellsort

Slide G.

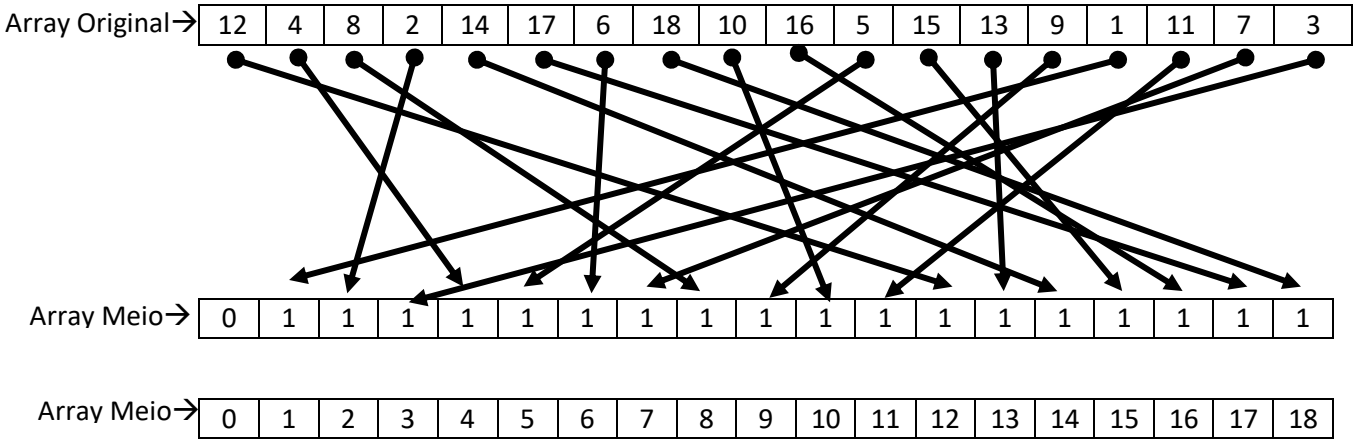
1)Em nosso exemplo, o algoritmo terminou sua execução?



Não, pois é necessário atualizar os elementos do array de contagem



2)Seja o array de entrada abaixo, quais serão os valores contidos no array de contagem antes e depois de copiarmos os elementos da entrada para a saída?



3) O Counting Sort pode ser aplicado adequadamente na ordenação de strings e números reais?

Resp: Não, no caso dos números reais temos infinitos números no intervalo 0 – 1 e como ele é indicado para manipulação de Inteiros não funcionaria em Reais. Porém é possível usar em uma string usando os valores da tabela Ascii como referência mas não seria funcional.

4) Nosso dinheiro é um número real. Conseguimos utilizar adequadamente o Counting Sort para ordenar valores financeiros?

Resp: Sim, pois apesar de Dinheiro estar dentro dos conjuntos reais sabemos que os centavos são finitos de até 100 unidades então basta multiplicar por 100 os valores, logo o 00,01 ocuparia a 1 posição, 00,10 a 10 posição 1,00 a 100 posição:

1)Mostre todas as comparações e movimentações do algoritmo anterior para o array abaixo:

12	4	8	2	14	17	6	18	10	16	5	15	13	9	1	11	7	3

Passos Básicos

Before Shell Sort
Vector {12,4,8,2,14,17,6,18,10,16,15,5,13,9,1,11,7,3}
Number case
Vector {0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
After Shell Sort
Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}

Todos os passos

Before Shell Sort
Vector {12,4,8,2,14,17,6,18,10,16,15,5,13,9,1,11,7,3}
Number -->Vector {0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
Ordenado-->Vector {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,0,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,0,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,0,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,0,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,0,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,0,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,0,0,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,0,0,0,0,0,0} *
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,0,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,0,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,0,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,0}
Ordenado-->Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}
After Shell Sort
Vector {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}