

1 Counting Sort (++++)



(++++)

Counting sort é um algoritmo de ordenação estável cuja complexidade é $O(n)$. As chaves podem tomar valores entre 0 e $M-1$. Se existirem k_0 chaves com valor 0, então ocupam as primeiras k_0 posições do vetor final: de 0 a k_0-1 .

O procedimento para implementação do Counting Sort segue os seguintes passos:

1. Cria-se um vetor $vCount[M+1]$ e $vOrd[N-1]$, onde N é a quantidade de elementos a serem ordenados e M é o maior valor entre os elementos a serem ordenados.
2. Inicializa-se todas as posições de $vCount$ com 0.
3. Percorre-se o vetor v e, para cada posição i de v faz-se $vCount[v[i]]++$, o que faz com que, no final, cada posição i de $vCount$ contem a quantidade de vezes que a chave i aparece em V .
4. Acumula-se em cada elemento de $vCount$ o elemento somado ao elemento anterior, desta forma, $vCount[i]$ indica a posição-1 ordenada do primeiro elemento de chave i .
5. Guarda-se em $vOrd$ os valores de V ordenados de acordo com $vOrd[vCount[v[i]-1]=V[i]$. E decrementa-se $vCount[v[i]]$ de uma unidade.
6. Copia-se $vOrd$ para v .

Esta implementação tem a desvantagem de precisar de vetores auxiliares. O Counting Sort ordena exclusivamente números inteiros pelo fato de seus valores servirem como índices no vetor de contagem.

Entrada

O programa possui vários casos de testes. A primeira de cada caso contém um inteiro N , $1 < N \leq 10000$, representando o tamanho do vetor. A segunda linha conterá N inteiros entre 0 e 1000, representando os N elementos do vetor. A entrada termina quando $N=0$.

Saída

O programa gera uma linha de saída para cada entrada, contendo os valores recebidos na entrada ordenados de acordo com o Counting Sort. Entre cada valor há um espaço em branco. Antes do primeiro valor não deve-se imprimir nada e após ao último valor deve-se apenas quebrar uma linha.

Exemplo

Entrada
10
6 13 7 3 13 6 14 3 14 9
5
9 8 7 6 5
8
0 1 2 3 4 5 6 7
0
Saída
3 3 6 6 7 9 13 13 14 14
5 6 7 8 9
0 1 2 3 4 5 6 7