

# SCC0201 - Introdução à Ciência de Computação II

## Lista de exercícios para aula

**Professor:** Diego Raphael Amancio

**Estagiárias PAE:**

Ana Caroline Medeiros Brito

### 1 *Bucket sort*

O bucket sort é um algoritmo de ordenação que divide valores a serem ordenados em buckets, isto é, em “baldes”. Há mais diferentes critérios de agrupamentos a fim de gerar os buckets. Uma forma usual é a partir de intervalos. Por exemplo, uma vez que sabemos o valor máximo  $M$  que um elemento pode assumir, então podemos considerar intervalos de tamanho  $M/N$ , onde  $N$  é o total de buckets.

A primeira parte do processo de ordenação é inserir os elementos em seus respectivos buckets conforme a regra pré-definida. Em seguida, em cada bucket, aplica-se um algoritmo de ordenação (bubble sort, insertion sort, etc). Por fim, os buckets são unidos em um único vetor, a fim de gerar a saída ordenada.

Nesse exercício, o critério para gerar os buckets será um pouco diferente. Para cada elemento, deve ser contada a quantidade de bits iguais a 1 em sua representação binária. Por exemplo, para o valor  $5_{10}$  ( $101_2$  na base binária) o número de bits iguais a 1 é 2. Assim, o valor  $5_{10}$  deve ser inserido no bucket 2.

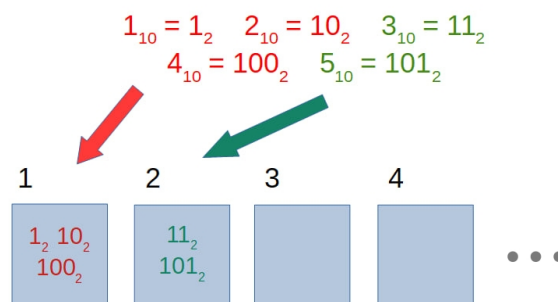


Figure 1: Exemplo ilustrado para os valores  $1_{10}$ ,  $2_{10}$ ,  $3_{10}$ ,  $4_{10}$  e  $5_{10}$ .

### Entrada

A primeira linha contém um único número inteiro  $N$ , indicando o número de inteiros que serão informados a seguir, a próxima linha contém  $N$  inteiros separados por espaço.

### Saída

A saída deve conter o número de linhas igual ao número de bucket distintos. Se um bucket permanecer vazio, ele não deve aparecer na saída. Imprima o conteúdo de cada bucket em uma nova linha. Os valores devem ser impressos de forma ordenada, do primeiro ao último bucket. Por fim, em uma nova linha, imprima todos os valores ordenados.

### Restrições

$$1 \leq N \leq 10^3$$

$$1 \leq A_i \leq 10^3$$

Nota: é garantido que cada elemento do vetor seja único.

### Exemplo

Entrada	Saída
5	1 2 4
1 2 3 4 5	3 5
	1 2 3 4 5

## 2 Ordenação números negativos

A implementação do algoritmo `radixSort(a,k)` funciona quando o vetor de entrada  $A$  contém apenas números inteiros não-negativos. Estenda esta implementação para que ela também funcione corretamente quando o vetor  $A$  contiver números inteiros negativos e não-negativos.

### Entrada

A primeira linha da entrada contém um inteiro que denota o número de elementos no vetor  $A$  ser ordenado. A linha a seguir contém  $N$  inteiros separados por espaço.

### Saída

Um linha com os  $N$  valores do vetor  $A$  ordenados.

### Exemplo

Entrada	Saída
5	-10 -1 3 50 99
3 -1 99 -10 50	