

# Cálculo da complexidade do RadixSort

Allan Cordeiro Rocha de Araújo.

<sup>1</sup>Centro de ciência e tecnologia– Universidade Federal de Roraima (UFRR)  
Boa Vista– RR – Brasil

[allanps32008@gmail.com](mailto:allanps32008@gmail.com)

**Resumo.** *Será calculado a complexidade do algoritmo de ordenação RadixSort.*

## 1. RadixSort

Ele usa uma ideia parecida com a tabela hash, que é atribuir chaves únicas aos seus elementos e ordenar baseado nessas chaves, ele usa da técnica de bits menos e mais significativos para tal, começando de dígitos menos significativos até os mais significativos.

## 2. Cálculo da complexidade

A complexidade do RadixSort varia de código para código, pois ela depende do algoritmo usado para ordenar cada dígito. Para o nosso cálculo usaremos o BucketSort para essa tarefa de ordenação dos dígitos, é aceitável pois sua complexidade é  $O(n+k)$ , como será mostrado a seguir.

BucketSort, pseudocódigo:

```
function bucket_sort(int A[], n, k)
begin
  crie o vetor de baldes (listas) B de tamanho k

  for i ← 1 to n do
    insira A[i] no seu balde B[msbits(A[i])] na ordem

  for i ← 1 to k do
    concatene os baldes B[i]
end
```

E o cálculo da complexidade do BucketSort:

Handwritten calculation of BucketSort complexity on lined paper. The text 'Bucket Sort' is written at the top. Below it, the formula  $\sum_{i=1}^m 1 + \sum_{i=1}^k 1 = (m-1+1) + (k-1+1) = m+k$  is written. At the bottom, the complexity is given as  $O(m+k)$ .

RadixSort, pseudocódigo:

```
function radix_sort(int A[], n)
begin
  for i ← 1 to n do
    ordene os elementos de A pelo i-ésimo dígito usando
    um metodo estável
  end
```

Ali onde tem “ordene os elementos...” é onde se aplica o algoritmo específico para a ordenação de dígitos, no nosso caso o BucketSort, que já está calculada a complexidade que é  $O(n+k)$ .

Cáculo da complexidade do RadixSort:

Handwritten calculation of RadixSort complexity on lined paper. The text 'Radix Sort' is written at the top. Below it, the formula  $\sum_{i=1}^p (m+k) = p(m+k)$  is written. At the bottom, the complexity is given as  $O(p(m+k))$ .

Assim a complexidade do RadixSort é  $O(p(n+k))$ .

## **Referências**

"MO417 - Complexidade de Algoritmos I"; Cid Carvalho de Souza, Cândida Nunes da Silva Orlando Lee. Disponível em

<<http://www.ic.unicamp.br/~zanoni/mo417/2011/aulas/handout/05-ordenacao-linear.pdf>>. Acesso em 12 de julho de 2018.

"Aula 10 – Métodos de Ordenação de Complexidade Linear"; Projeto de Análise de Algoritmos. Disponível em

<[http://edirlei.3dgb.com.br/aulas/paa\\_2016\\_1/PAA\\_Aula\\_10\\_Ordenacao\\_Complexidade\\_Linear\\_2016.pdf](http://edirlei.3dgb.com.br/aulas/paa_2016_1/PAA_Aula_10_Ordenacao_Complexidade_Linear_2016.pdf)>. Acesso em 12 de julho de 2018.