

Método de Ordenação — QuickSort

Código e Análise de Complexidade

Larissa; Mauricio Monteiro; Paulo Henrique

UFMG — ICEx
Departamento de Ciência da Computação — DCC
Ambientes de Computação

Visão geral da apresentação

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

- 1 Introdução
- 2 Código em C
- 3 Código em Python
- 4 Lógica do Algoritmo
- 5 Passo a Passo
- 6 Análise de Complexidade

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

O trabalho tem como objetivo:

- traduzir o código escrito na *linguagem C* para a *linguagem Python*;
- compreender o funcionamento do algoritmo proposto;
- explicar a **Análise de Complexidade** do código.

Código em C

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

```
11 #define MAXTAM 20
12
13 typedef long TipoChave;
14
15 typedef struct TipoItem
16 {
17     TipoChave Chave;
18 } TipoItem;
19
20 typedef int TipoIndice;
21 typedef TipoItem TipoVetor[MAXTAM + 1];
22
23 void Particao(TipoIndice Esq, TipoIndice Dir, TipoIndice *i, TipoIndice *j, TipoItem *A)
24 {
25     TipoItem x, w;
26
27     *i = Esq;
28     *j = Dir;
29
30     x = A[( *i + *j ) / 2]; /* obtem o pivo x */
31
32     do
33     {
34         while (x.Chave > A[*i].Chave)
35         {
36             (*i)++;
37         }
38
39         while (x.Chave < A[*j].Chave)
40         {
41             (*j)--;
42         }
43
44         if (*i <= *j)
45         {
46             w = A[*i];
47             A[*i] = A[*j];
48             A[*j] = w;
49
50             (*i)++;
51             (*j)--;
52         }
53     } while (*i <= *j);
54 }
55
```

Trabalho

Introdução

Código em C

Código em Python

Lógica do Algoritmo

Passo a Passo

Análise de Complexi- dade

```
56
57 void Ordena(TipoIndice Esq, TipoIndice Dir, TipoItem *A)
58 {
59
60     TipoIndice i = 0, j = 0;
61
62     Particao(Esq, Dir, &i, &j, A);
63
64     if (Esq < j)
65     {
66         Ordena(Esq, j, A);
67     }
68
69     if (i < Dir)
70     {
71         Ordena(i, Dir, A);
72     }
73 }
74
75 void QuickSort(TipoItem *A, TipoIndice n)
76 {
77     Ordena(0, n - 1, A);
78 }
79
80 TipoVetor A;
81
```

Código em Python

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

```
24 def Ordena(l, r, A):
25
26     i, j = Particao(l, r, A)
27
28     if l < j:
29         Ordena(l, j, A)
30
31     if i < r:
32         Ordena(i, r, A)
33
34     pass
35
36 def QuickSort(A):
37     n = len(A) - 1
38     Ordena(0, n, A)
39     pass
40
```

Código em Python

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

```
1 def Particao (i, j, A):
2
3     x = A[(i + j) // 2] # Obtem o pivo x
4
5     ok = True
6     while ok:
7
8         while x > A[i]:
9             i += 1
10
11        while x < A[j]:
12            j -= 1
13
14        if i <= j:
15            A[i], A[j] = A[j], A[i] # swap(A[i], A[j])
16
17            i += 1
18            j -= 1
19
20        ok = True if i <= j else False
21
22    return i, j
23
```

Lógica do Algoritmo

Trabalho

Introdução

Código em C

Código em Python

Lógica do Algoritmo

Passo a Passo

Análise de Complexi- dade

- A idéia básica é a de partir o problema de ordenar um conjunto $A = \{a_0, a_1, \dots, a_{n-1}, a_n\}$ com n itens em **dois problemas menores**.
- Os problemas menores são ordenados independentemente e depois os resultados são combinados para produzir a solução do problema maior.

Lógica do Algoritmo

Trabalho

Introdução

Código em C

Código em Python

Lógica do Algoritmo

Passo a Passo

Análise de Complexidade

- A parte mais delicada deste método é relativa ao procedimento partição, o qual tem que reorganizar o vetor $A = \{a_{\text{esq}}, \dots, a_{\text{dir}}\}$ através da escolha arbitrária de um item x do vetor chamado **pivô**.
- Ao final o vetor A estará particionado em uma **parte esquerda** com chaves **menores ou iguais** a x e uma **parte direita** com chaves **maiores ou iguais** a x .

Trabalho

Introdução

Código em C

Código em Python

Lógica do Algoritmo

Passo a Passo

Análise de Complexi- dade

Este comportamento pode ser descrito pelo seguinte algoritmo:

- 1 escolha arbitrariamente um item do vetor e o coloque em x ;
- 2 percorra o vetor a partir da esquerda até que um item $a_i > x$ é encontrado.
- 3 da mesma forma percorra o vetor a partir da direita até que um item $a_j \leq x$ é encontrado;
- 4 como os dois itens A_i e A_j estão fora de lugar no vetor final então **troque-os de lugar**;
- 5 continue este processo até que os apontamentos i e j se **cruzem** em algum ponto do vetor.

Trabalho

Introdução

Código em C

Código em Python

Lógica do Algoritmo

Passo a Passo

Análise de Complexi- dade

Ao final o vetor $A = \{a_{\text{esq}}, \dots, a_{\text{dir}}\}$ está parcionado de tal forma que:

- os itens em $a_{\text{esq}}, a_{\text{esq}+1}, \dots, a_j$ são **menores** ou iguais a x ,
- os itens em $a_j, a_{j+1}, \dots, a_{\text{dir}}$ são **maiores** ou iguais a x .

Passo a Passo

Trabalho

0	1	2	3	4	5	6
1	8	3	9	4	5	7

L	0	1	2	3	4	5	R	6
	1	8	3	9	4	5		7

$i \rightarrow j$

0	1	2	3	4	5	6
1	8	3	7	4	5	9

$i \quad j$

0	1	2	3	4	5	6
1	8	3	7	4	5	9

$i \rightarrow j$

0	1	2	3	4	5	6
1	8	3	7	4	5	9

Ordena Iteração 1

Passo a Passo

Trabalho

Introdução

Código em C

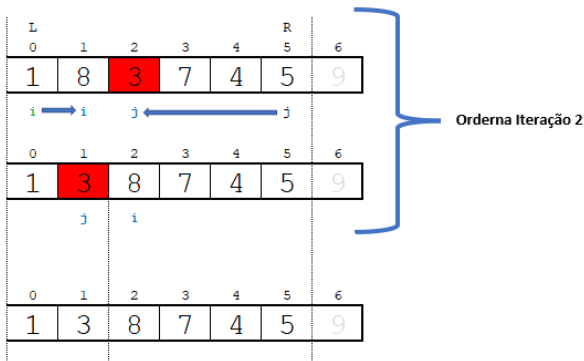
Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

0	1	2	3	4	5	6
1	8	3	7	4	5	9



Passo a Passo

Trabalho

Introdução

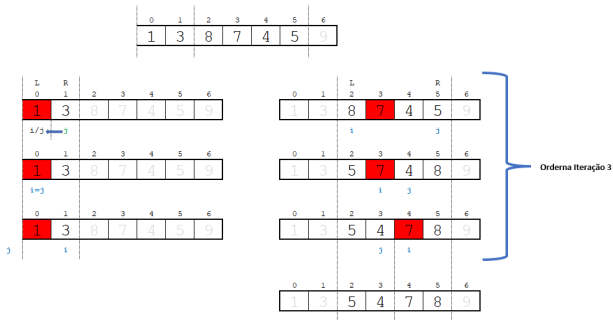
Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade



Passo a Passo

Trabalho

Introdução

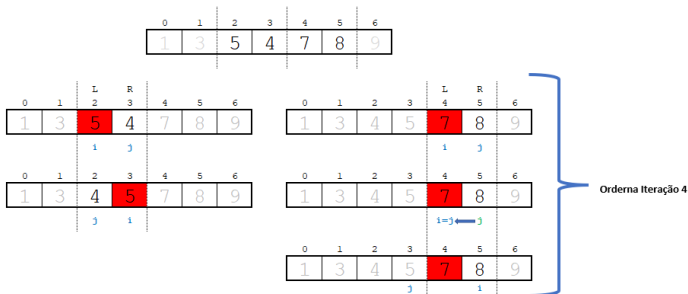
Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade



Passo a Passo

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

Animação do exemplo proposto:

Trabalho

Introdução

Código em C

Código em
Python

Lógica do
Algoritmo

Passo a Passo

Análise de
Complexi-
dade

nossa introdução



Obrigado.

Perguntas?