

Relatorio

Karine Piacentini Coelho da Costa¹

March 12, 2019

¹karinepcdc@ufrn.br

Contents

1	Introdução	2
2	Métodos	3
2.1	Caracterização técnica do computador	3
2.2	Algoritmos	3
2.2.1	Busca linear	3
2.2.2	Busca binária	4
2.2.3	Busca ternária	5
2.2.4	<i>Jump search</i>	6
2.2.5	Busca de Fibonacci	6
3	Resultados	7
3.1	7

Chapter 1

Introdução

Chapter 2

Métodos

2.1 Caracterização técnica do computador

2.2 Algoritmos

2.2.1 Busca linear

Algoritmo 1: Busca linear

Entrada: Vetor V , chave k e limites de busca esquerdo l e direito r (inclusive).

Saída: Índice da ocorrência de k em V ; ou -1 caso não exista k em V .

/ Precondição: $l \leq r$; $l, r \geq 0$; V em ordem crescente. */*

```
1 Função buscaLin( $V$ : arranjo de inteiros;  $l$ : inteiro;  $r$ : inteiro;  $k$ :  
  inteiro): inteiro  
2   var  $i$ : inteiro  
3   enquanto  $i \leftarrow l$  até  $r$  faça  
4       se  $V[i] == k$  então  
5           retorna  $i$   
6       fim  
7   fim  
8   retorna  $-1$   
9 fim
```

2.2.2 Busca binária

Algoritmo 2: Busca binária iterativa

Entrada: Vetor V , chave k e limites de busca esquerdo l e direito r (inclusive).
Saída: Índice da ocorrência de k em V ; ou -1 caso não exista k em V .
 /* Precondição: $l \leq r$; $l, r \geq 0$; V em ordem crescente. */

```

1  Função buscaBin( $V$ : arranjo de inteiros;  $l$ : inteiro;  $r$ : inteiro;  $k$ :
    inteiro): inteiro
2      var  $m$ : inteiro
3      enquanto  $r \geq l$  faça
4           $m \leftarrow (l + r)/2$ 
5          se  $k == V[m]$  então
6              retorna  $m$ 
7          senão se  $k < V[m]$  então
8               $r \leftarrow m - 1$ 
9          senão
10              $l \leftarrow m + 1$ 
11         fim
12     fim
13     retorna  $-1$ 
14 fim
```

Algoritmo 3: Busca binária recursiva

Entrada: Vetor V , chave k e limites de busca esquerdo l e direito r (inclusive).
Saída: Índice da ocorrência de k em V ; ou -1 caso não exista k em V .
 /* Precondição: $l \leq r$; $l, r \geq 0$; V em ordem crescente. */

```

1  Função buscaBin_rec( $V$ : arranjo de inteiros;  $l$ : inteiro;  $r$ : inteiro;  $k$ :
    inteiro): inteiro
2      var  $m$ : inteiro
3      se  $r < l$  então
4          retorna  $-1$ 
5      senão
6           $m \leftarrow (l + r)/2$ 
7          se  $k == V[m]$  então
8              retorna  $m$ 
9          senão se  $k < V[m]$  então
10             buscaBin_rec( $V, l, m - 1, k$ )
11          senão
12             buscaBin_rec( $V, m + 1, r, k$ )
13      fim
14  fim
15 fim
```

2.2.3 Busca ternária

Algoritmo 4: Busca ternária iterativa

Entrada: Vetor V , chave k e limites de busca esquerdo l e direito r (inclusive).

Saída: Índice da ocorrência de k em V ; ou -1 caso não exista k em V .

/ Precondição: $l \leq r$; $l, r \geq 0$; V em ordem crescente. */*

```
1 Função buscaTer( $V$ : arranjo de inteiros;  $l$ : inteiro;  $r$ : inteiro;  $k$ :  
   inteiro): inteiro  
2   var  $m$ : inteiro  
3   enquanto  $r \geq l$  faça  
4      $m \leftarrow (l + r)/2$   
5     se  $k == V[m]$  então  
6       retorna  $m$   
7     senão se  $k < V[m]$  então  
8        $r \leftarrow m - 1$   
9     senão  
10       $l \leftarrow m + 1$   
11    fim  
12  fim  
13  retorna  $-1$   
14 fim
```

2.2.4 *Jump search*

Algoritmo 5: *Jump search*

Entrada: Vetor V , chave k e limites de busca esquerdo l e direito r (inclusive).**Saída:** Índice da ocorrência de k em V ; ou -1 caso não exista k em V ./* Precondição: $l \leq r$; $l, r \geq 0$; V em ordem crescente. */

```

1  Função buscaJump( $V$ : arranjo de inteiros;  $l$ : inteiro;  $r$ : inteiro;  $k$ :
   inteiro): inteiro
2  |   var  $m$ : inteiro
3  |   var  $p$ : inteiro /* tamanho do salto */
4  |
5  |    $p \leftarrow \sqrt{r - l + 1}$ 
6  |    $m \leftarrow l + p$ 
7  |   enquanto  $m \leq r$  faça
8  |       |   se  $k == V[m]$  então
9  |           |   retorna  $m$ 
10 |       |   senão se  $k < V[m]$  então
11 |           |   retorna buscaLin( $V, m - p, m - 1, k$ )
12 |       |   fim
13 |       |    $m = m + p$ 
14 |   fim
15 |   se  $m > r$  e  $V[r] > k$  então
16 |       |   retorna buscaLin( $V, m - p, r, k$ )
17 |   fim
18 |   retorna  $-1$ 
19 fim

```

2.2.5 Busca de Fibonacci

Chapter 3

Resultados

3.1