

ED1 - códigos

GABRIEL LAZARI TREVISANI

September 2024

1 Ordenação

1.1 Algoritmos $O(n^2)$

Algorithm 1 Bubble sort

```
1: procedure BUBBLE SORT( $arr, n$ )
2:   for  $i \leftarrow 0$  to  $n - 2$  do
3:     for  $j \leftarrow 0$  to  $n - 1 - i$  do           ▷ move maior elem para ult. pos.
4:       if  $arr[j] > arr[j + 1]$  then
5:          $arr[j] \longleftrightarrow arr[j + 1]$ 
6:       end if
7:     end for
8:   end for
9: end procedure
```

Algorithm 2 Selection sort

```
1: procedure SELECTION SORT( $arr, n$ )
2:   for  $i \leftarrow 0$  to  $n - 1$  do
3:      $chave \leftarrow i$ 
4:     for  $j \leftarrow i + 1$  to  $n$  do           ▷ acha o menor elemento
5:       if  $arr[chave] > arr[j]$  then
6:          $chave \leftarrow j$ 
7:       end if
8:     end for
9:      $v[i] \longleftrightarrow v[chave]$        ▷ troca o menor elemento com a posição inicial
10:  end for
11: end procedure
```

Algorithm 3 Insertion sort

```
1: procedure INSERTION SORT( $arr, n$ )
2:   for  $i \leftarrow 1$  to  $n - 1$  do
3:      $chave \leftarrow i$ 
4:     for  $j \leftarrow i - 1$  to  $0$  do ▷ insere no conjunto ordenado
5:       if  $arr[chave] < arr[j]$  then
6:          $arr[chave] \longleftrightarrow arr[j]$ 
7:          $chave \leftarrow j$  ▷ troca o até chegar no lugar certo
8:       end if
9:     end for
10:  end for
11: end procedure
```

1.2 Algoritmos $O(n \cdot \log n)$

Algorithm 4 Quick sort

```
1: procedure QUICK SORT( $arr, a, b$ )
2:   if  $a < b$  then ▷ Só acaba se tiver 1 elemento pra dividir em 2
3:      $pos\_pivo \leftarrow Particiona(arr, a, b)$  ▷ Divisão e conquista
4:      $Quicksort(arr, a, chave)$ 
5:      $Quicksort(arr, chave + 1, b)$ 
6:   end if
7: end procedure
8:
9: procedure PARTICIONA( $arr, a, b$ ) ▷ Divide em  $\leq pivo$  e  $> pivo$ 
10:   $pivo \leftarrow arr[b]$  ▷ põe pivo no fim pra percorrer todo array
11:   $pos\_pivo \leftarrow a$  ▷ separa  $pivo$  e  $pos\_pivo$  ( $b \neq pos\_pivo$ )
12:  for  $i \leftarrow a$  to  $b - 1$  do ▷ pivo fica em elemento  $\leq$  até aparecer o  $1^\circ >$ 
13:    if  $arr[i] \leq pivo$  then
14:       $arr[i] \longleftrightarrow arr[pos\_pivo]$  ▷ troca  $1^\circ$  elemento  $>$  com o  $\leq$ 
15:       $pos\_pivo++$  ▷ Incrementa pra ficar no  $1^\circ$  elemento  $>$ 
16:    end if ▷ Fica sempre no  $1^\circ$  elemento  $>$  agora
17:  end for
18:  return  $pos\_pivo$ 
19: end procedure
20:
21: procedure RANDOM PARTICIONA( $arr, a, b$ )
22:   $pos\_pivo \leftarrow a + (rand() \% (b - a + 1))$  ▷ recebe valor aleatório entre a e b
23:   $arr[pos\_pivo] \longleftrightarrow arr[b]$  ▷ põe no final pra usar ele de pivo
24:  return  $Particiona(arr, a, b)$ 
25: end procedure
```

Algorithm 5 Merge Sort

```
procedure MERGE SORT(arr, a, b)
  if  $b > a$  then                                     ▷ Divide até ter vetor unitário
     $m \leftarrow \frac{a+b}{2}$ 
    MergeSort(arr, a, m)
    MergeSort(arr, m + 1, b)
  end if
  Merge(arr, a, b)                                     ▷ Funde 2 vetores já ordenados
end procedure
```

```
procedure MERGE(arr, a, b)
   $m \leftarrow \frac{a+b}{2}$ 
  Cria vetor right e left dividindo em  $m$  e preenche com as metades
   $i \leftarrow 0, j \leftarrow 0, k \leftarrow a$ 
  while  $i < m$  &&  $j < b$  do                           ▷ Põe o menor deles até um acabar
    if  $left[i] > right[j]$  then
       $arr[k++] \leftarrow right[j++]$ 
    else
       $arr[k++] \leftarrow left[i++]$ 
    end if
  end while                                             ▷ Falta colocar o que sobrou de um deles
  while  $i < m$  do
     $arr[k++] \leftarrow left[i++]$ 
  end while
  while  $j < b$  do
     $arr[k++] \leftarrow right[j++]$ 
  end while
  Desaloque right e left
end procedure
```

1.3 Algoritmo $O(\log n)$

Algorithm 6 Busca binária

```
1: procedure BUSCA BINÁRIA(arr, num, n)
2:    $inf \leftarrow 0$ 
3:    $sup \leftarrow n$ 
4:    $i \leftarrow \frac{inf+sup}{2}$ 
5:   while  $arr[i] \neq num$  do
6:     if  $arr[i] > num$  then
7:        $sup \leftarrow i$ 
8:     else
9:        $inf \leftarrow i$ 
10:    end if
11:     $i \leftarrow \frac{inf+sup}{2}$ 
12:  end while
13: end procedure
```
