

# Lista de Exercícios 4

Gustavo Higuchi

August 25, 2016

## Contents

<b>Exercício 1</b>	<b>2</b>
<b>Exercício 2</b>	<b>2</b>
<b>Exercício 3</b>	<b>3</b>

## Exercício 1

Informalmente, o custo do algoritmo ficaria  $\Theta(n^3)$ .

## Exercício 2

(a)

```
Input: Um vetor A[1..n]
Output: O vetor A ordenado
for  $i \leftarrow 0$  to  $n - 1$  do
     $m \leftarrow i$ 
    for  $j \leftarrow i + 1$  to  $n$  do
        if  $A[m] < A[j]$  then
             $m \leftarrow j$ 
        end
    end
     $aux \leftarrow A[i]$ 
     $A[i] \leftarrow A[m]$ 
     $A[m] \leftarrow aux$ 
end
```

**Algorithm 1:** Selection Sort

(b)

Como o menor elemento do vetor é sempre trocado com a primeira posição do vetor, quando o algoritmo chega no último elemento, todos os  $n - 1$  elementos anteriores são menores que o último elemento.

(c)

Para este algoritmo, o melhor caso é igual ao pior caso, e é exatamente  $(n - 1) *$

$$\sum_{i=1}^{n-1} (n - i)$$

(d)

Dado o tempo exato do exercício (c), assintoticamente falando o algoritmo roda em tempo  $\Theta(n^2)$ .

### Exercício 3

(a)

**Input:** Uma sequência de  $n$  números  $A = (a_1, a_2, \dots, a_n)$  e um valor  $v$   
**Output:** Um índice tal que  $v = A[i]$  ou um valor especial *NIL* se  $v$  não aparece em  $A$

```
for  $i \leftarrow 0$  to  $n$  do
    if  $A[i] = v$  then
        return  $v$ 
    end
end
return NIL
```

**Algorithm 2:** Busca Linear

(b)

(c)

Se o elemento pode estar distribuído estatisticamente semelhante, metade das vezes estará na metade inicial, na outra metade, estará na metade final. Daria uma média de  $\frac{n}{2}$  comparações

(d)

O pior caso seria se o elemento estivesse na última posição do vetor, fazendo  $n$  comparações.

(e)

Em ambos os casos, na notação assintótica, teremos um tempo de execução de  $\Theta(n)$ .