

Busca em Memória Primária

Letícia Rodrigues Bueno

Busca Sequencial

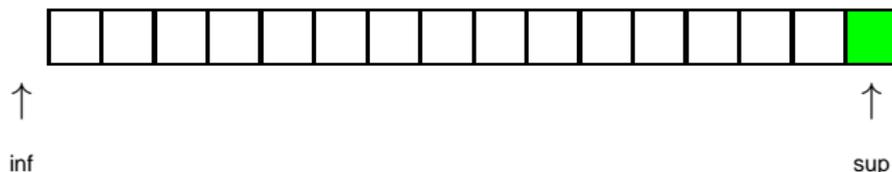
buscaSequencial(x, V)

- 1: $i \leftarrow 1$;
- 2: **enquanto** ($i \leq n$) **e** ($V[i] \neq x$) **faça**
- 3: $i \leftarrow i + 1$;
- 4: **se** $i > n$ **então** “Busca sem sucesso”
- 5: **senão** “Busca com sucesso”

```

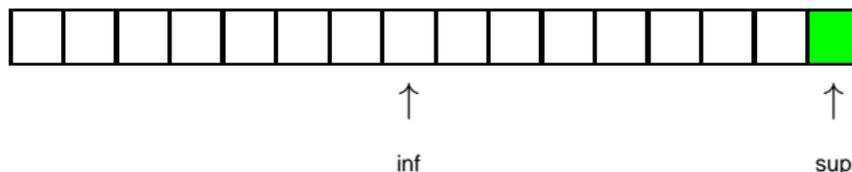
1: BuscaBinária(chave, lista[0 . . . n])
2: inf ← -1
3: sup ← n
4: enquanto inf < sup - 1 faça
5:   meio ←  $\lfloor \frac{\textit{inf} + \textit{sup}}{2} \rfloor$ 
6:   se chave ≤ lista[meio] então
7:     sup ← meio
8:   senão
9:     inf ← meio
10: se chave = lista[sup] então
11:   retorna lista[sup]
12: senão
13:   retorna elemento não encontrado

```



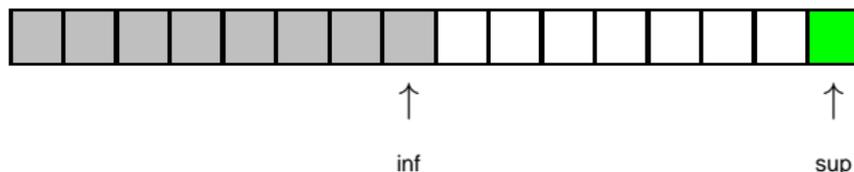
inicializ.

- 1: BuscaBinária(*chave*, *lista*[0 . . . *n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{i\text{nf}+s\text{up}}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



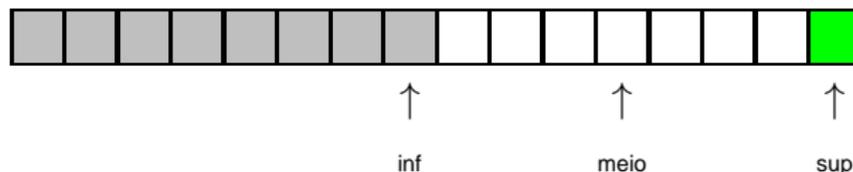
1a iter.

- 1: BuscaBinária(*chave*, *lista*[0 . . . *n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{inf+sup}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



2a iter.

- 1: BuscaBinária(*chave*, *lista*[0 . . . *n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{i\text{nf} + \text{sup}}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



2a iter.

- 1: BuscaBinária(*chave*, *lista*[0 . . . *n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{i\text{nf} + \text{sup}}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



3a iter.

- 1: BuscaBinária(*chave*, *lista*[0...*n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{inf+sup}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



4a iter.

- 1: BuscaBinária(*chave*, *lista*[0 . . . *n*])
- 2: *inf* \leftarrow -1
- 3: *sup* \leftarrow *n*
- 4: **enquanto** *inf* < *sup* - 1 **faça**
- 5: *meio* \leftarrow $\lfloor \frac{inf+sup}{2} \rfloor$
- 6: **se** *chave* \leq *lista*[*meio*] **então**
- 7: *sup* \leftarrow *meio*
- 8: **senão**
- 9: *inf* \leftarrow *meio*
- 10: **se** *chave* = *lista*[*sup*] **então**
- 11: **retorna** *lista*[*sup*]
- 12: **senão**
- 13: **retorna** elemento não encontrado



inf *sup*

5a iter.

Bibliografia

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. e STEIN, C.
Introduction to Algorithms, 3ª edição, MIT Press, 2009.

SZWARCFITER, J. L. e MARKENZON, L. Estruturas de Dados e seus Algoritmos, LTC, 1994.

ZIVIANI, N. Projeto de Algoritmos: com implementações em Pascal e C, 2ª edição, Cengage Learning, 2009.