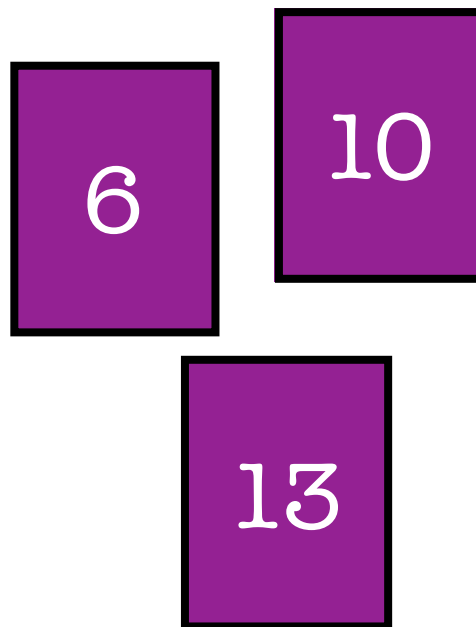


# containers

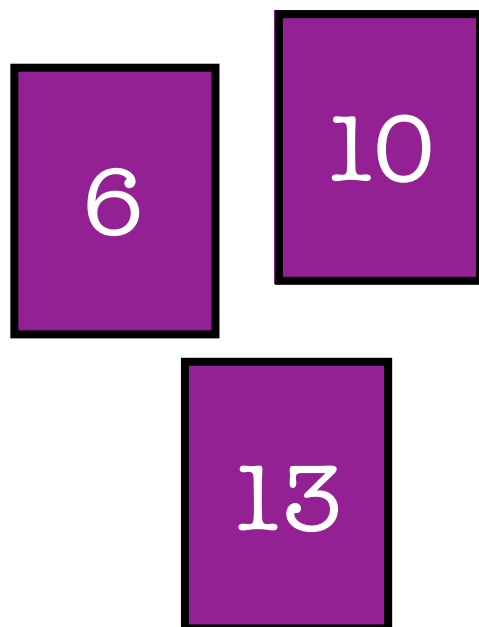
(estruturas & TADs)

TADs



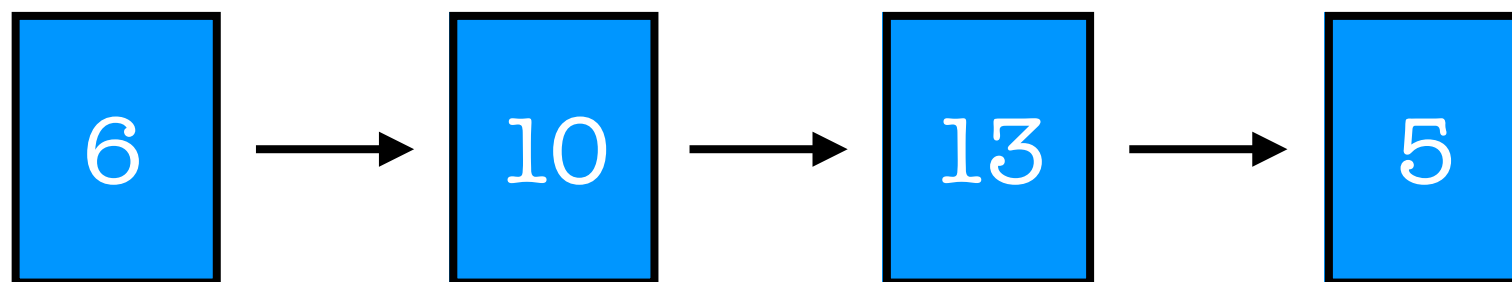
(não-relacionais)

TADs

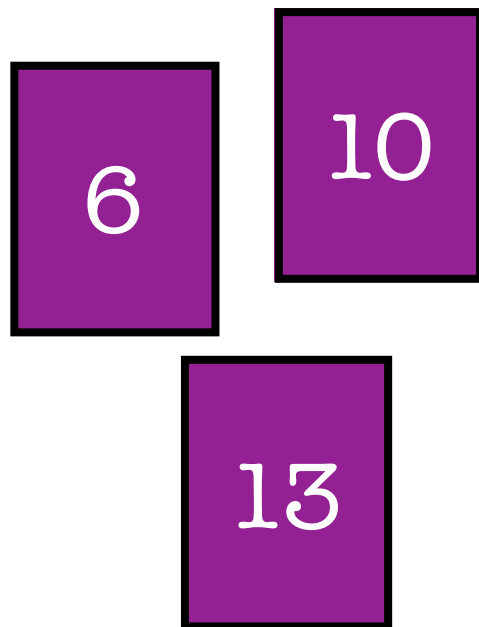


(não-relacionais)

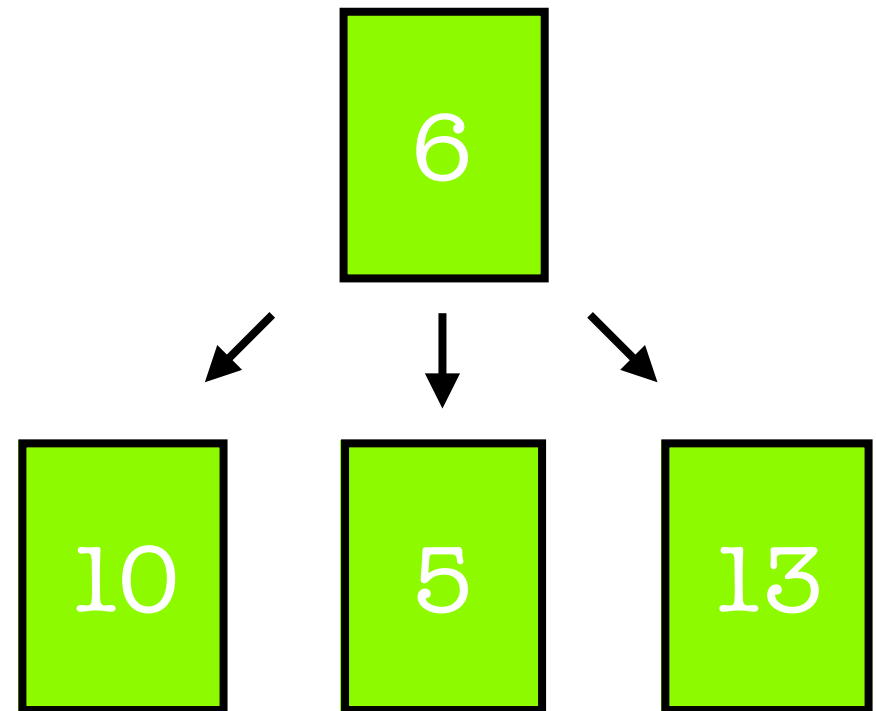
TADs



(lineares)

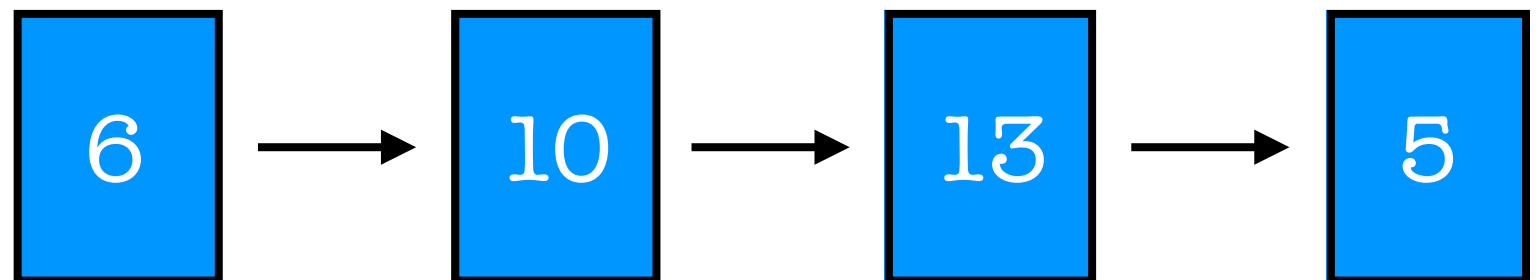


(não-relacionais)

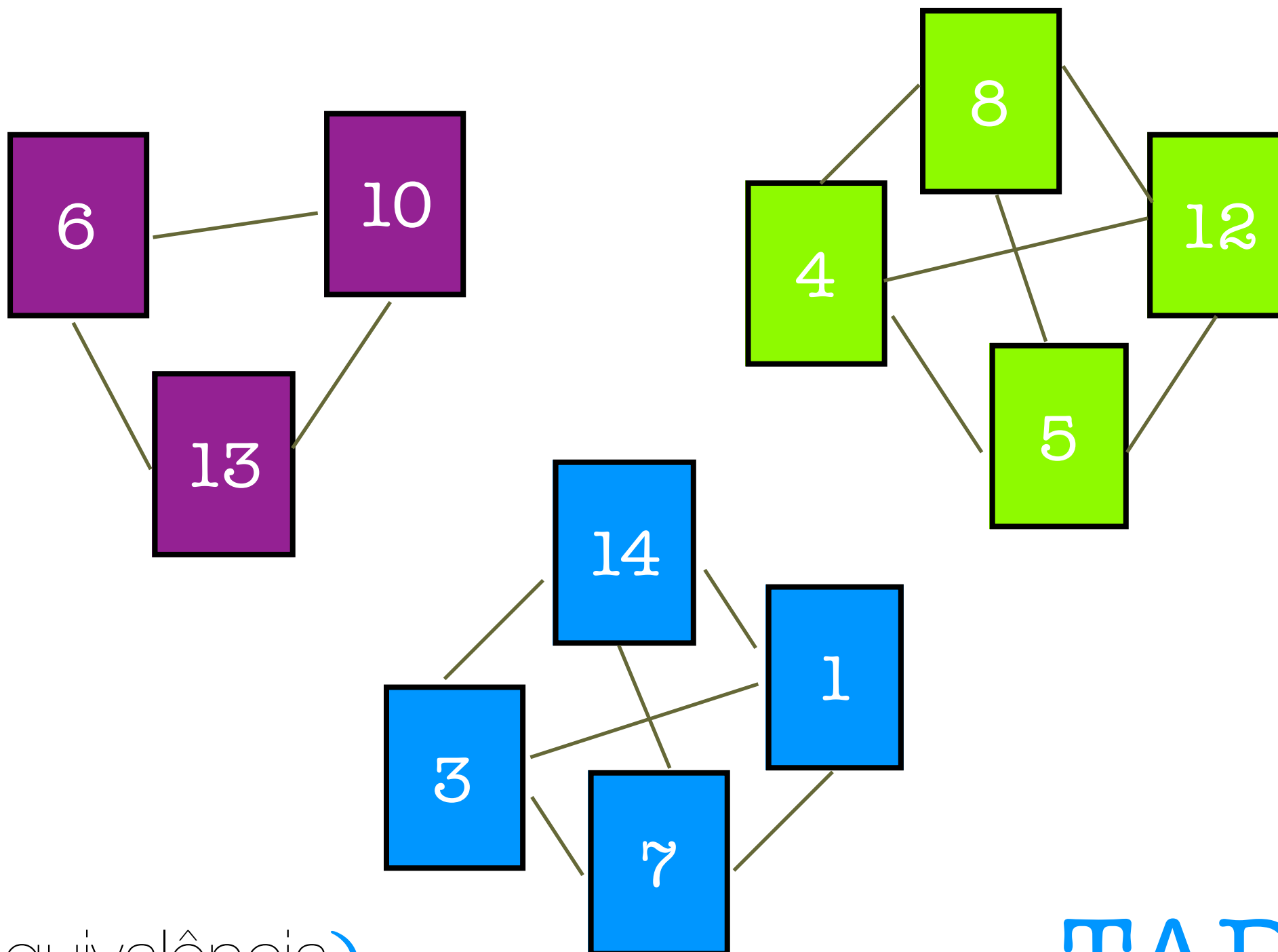


(hierárquicos)

TADs

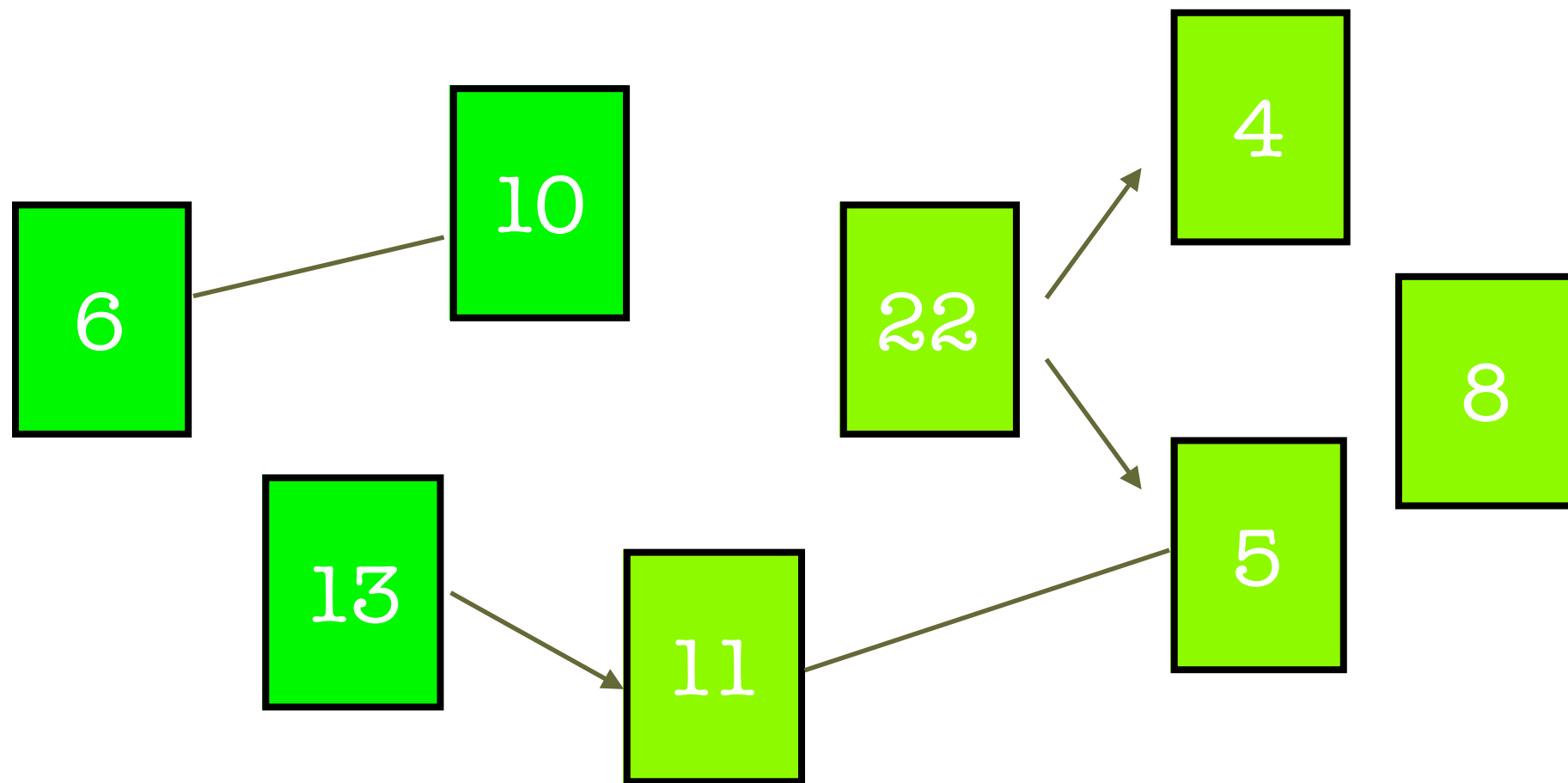


(lineares)



(equivalência)

TADs



(adjacência)

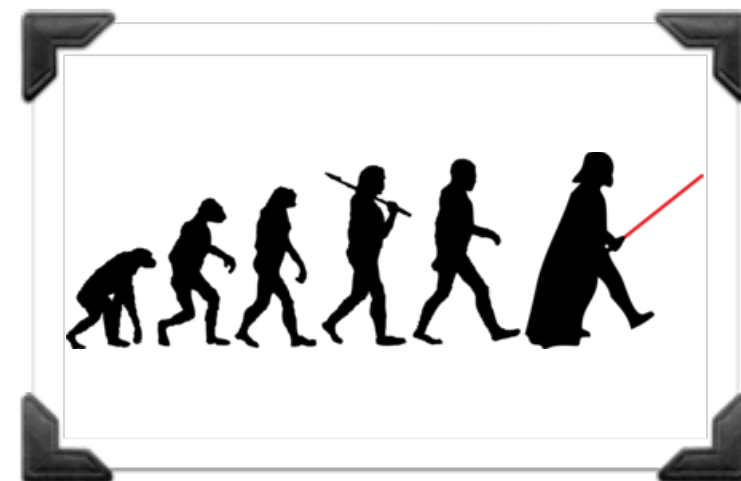
TADs



TADs  
(lineares)



lista



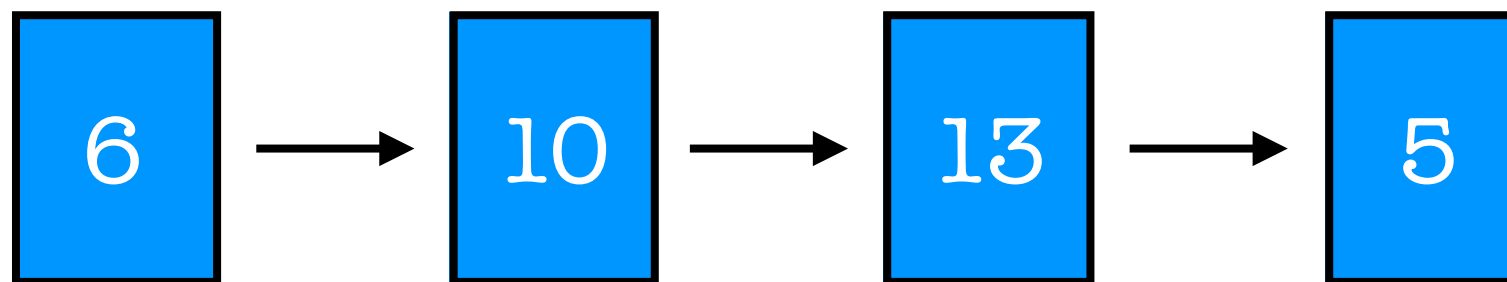
sequência

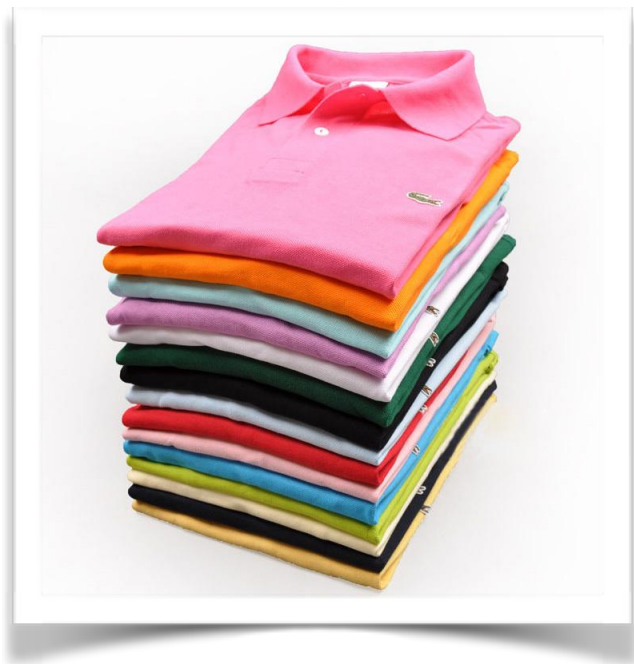
➡ Ordem de **inserção**

---

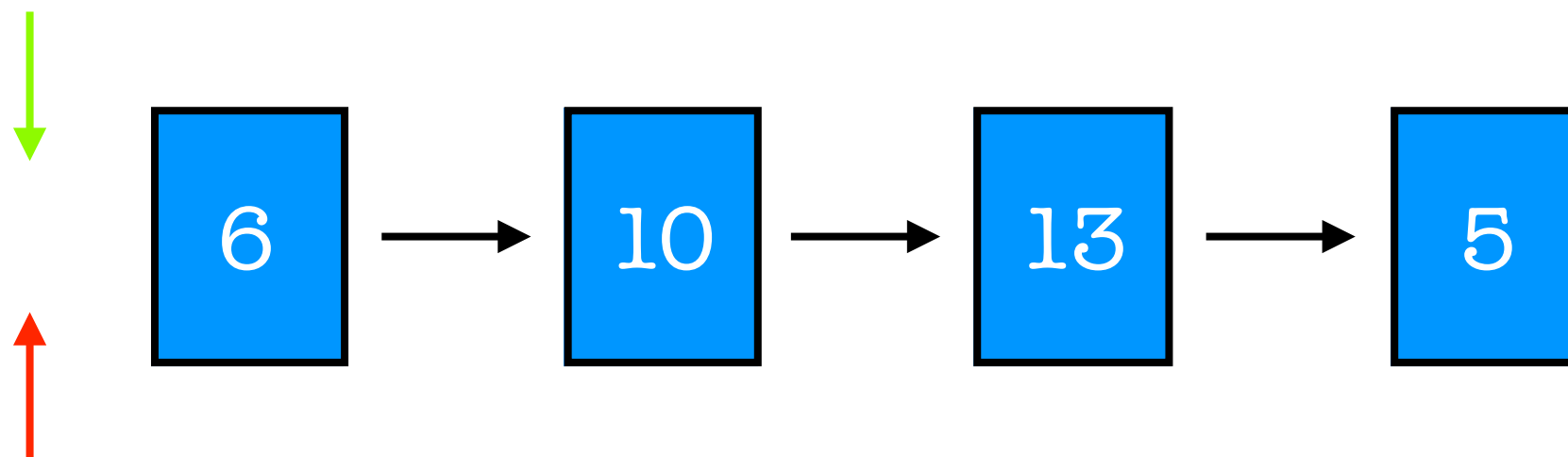
➡ Ordem de **remoção**

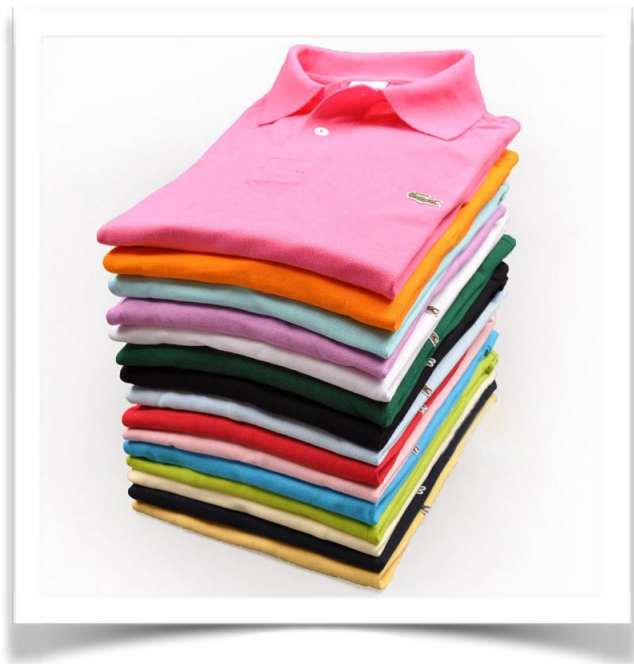
TADs  
(lineares)



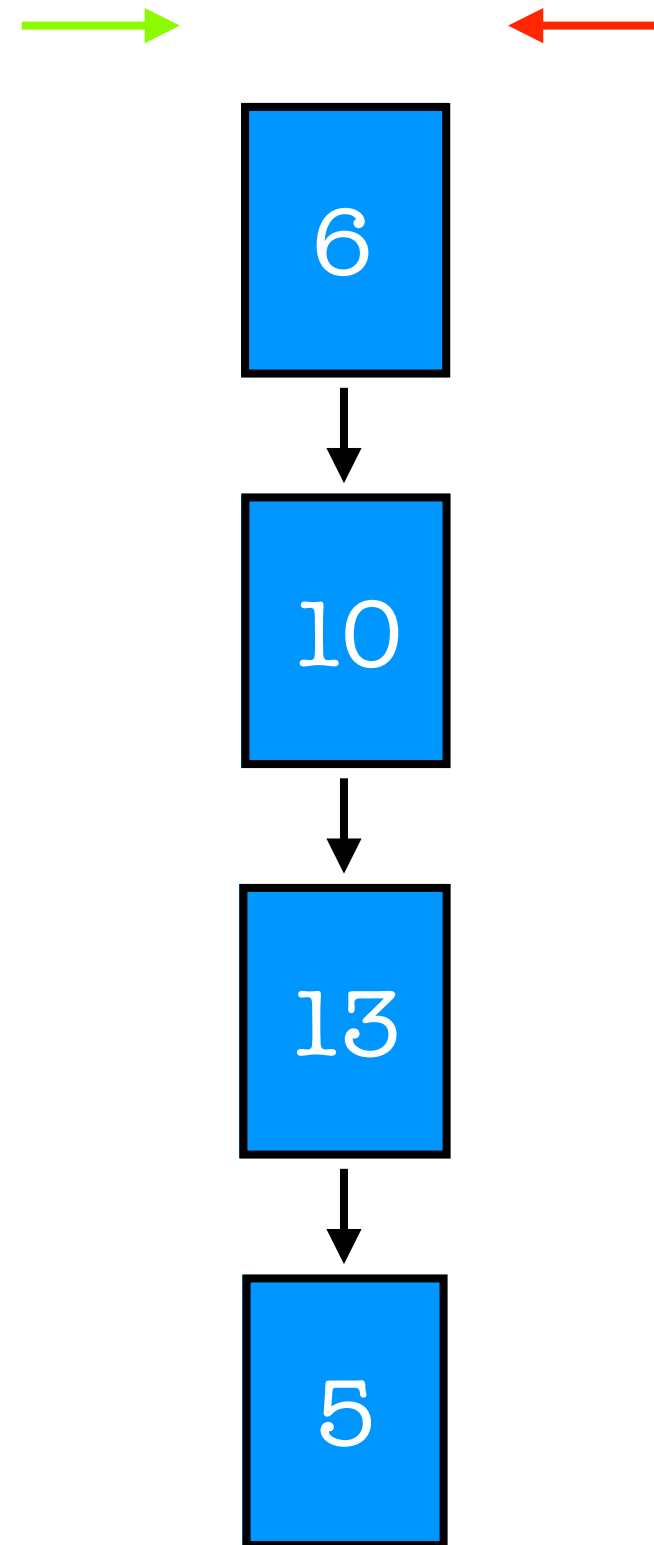


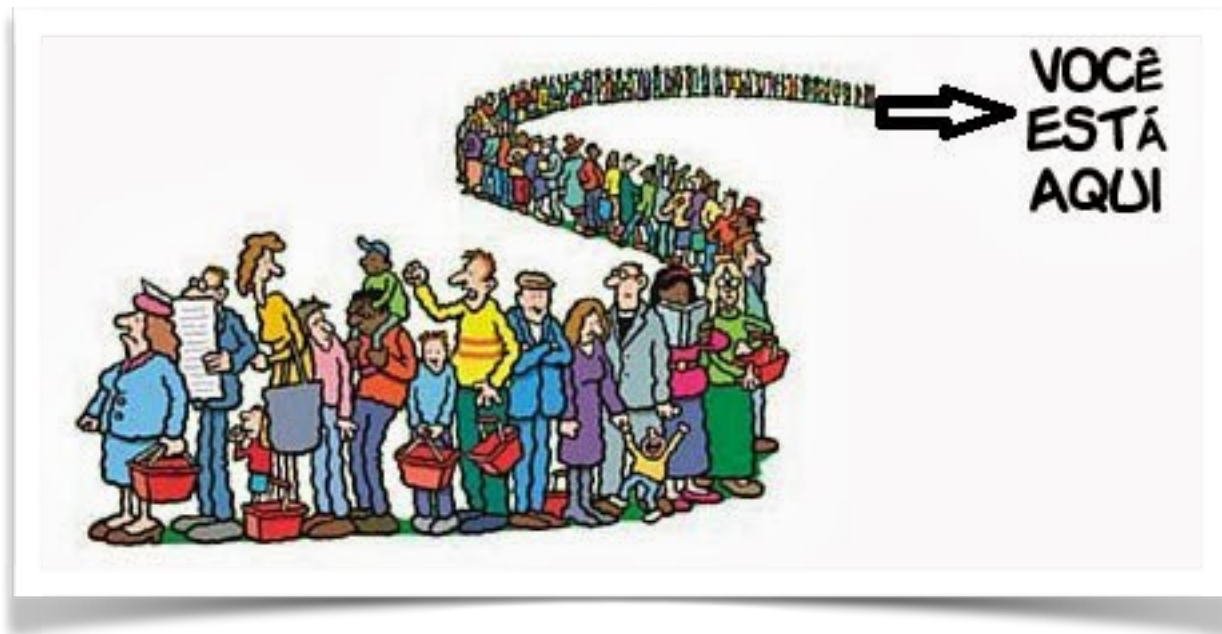
TAD  
(pilha)



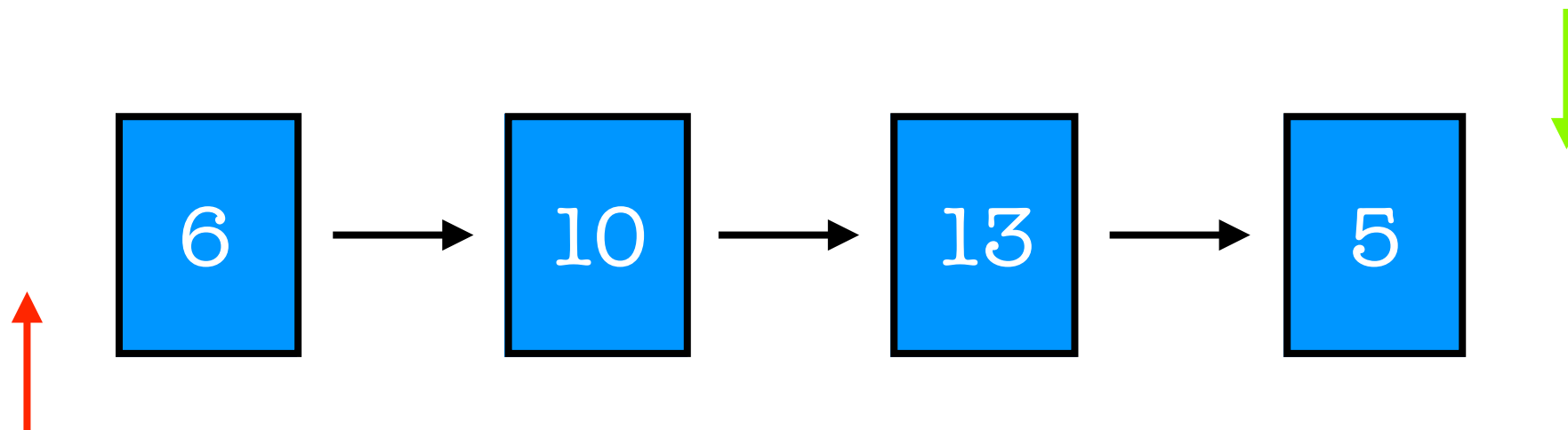


TAD  
(pilha)





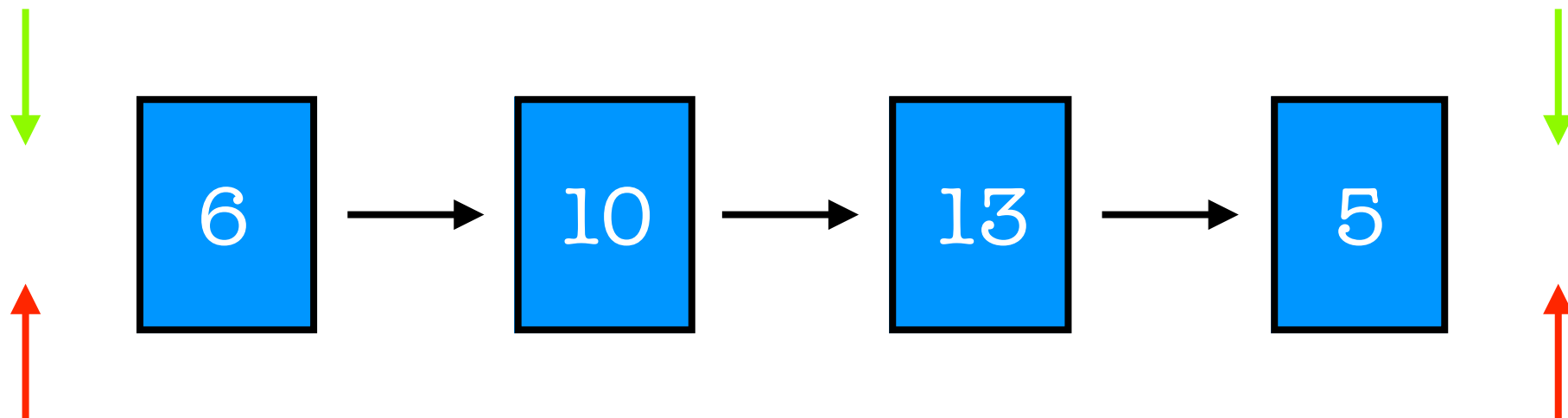
TAD  
(fila)



Políticos  
entram  
aqui



TAD  
(deque)



lista

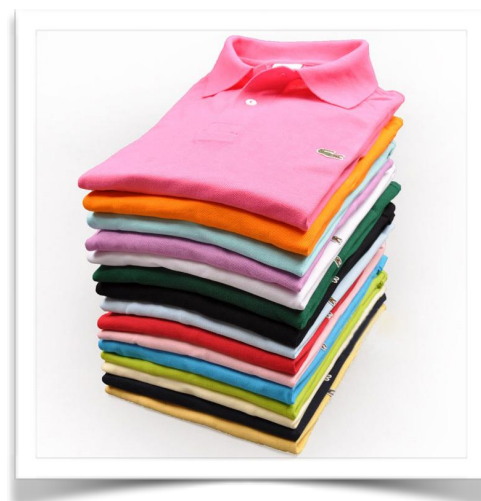


Políticos  
entram  
aqui



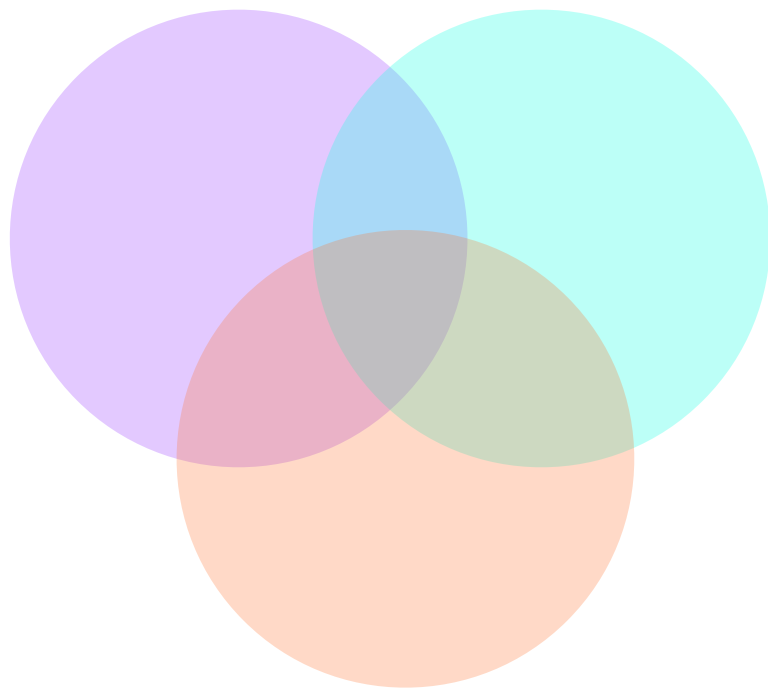
deque

pilha



fila

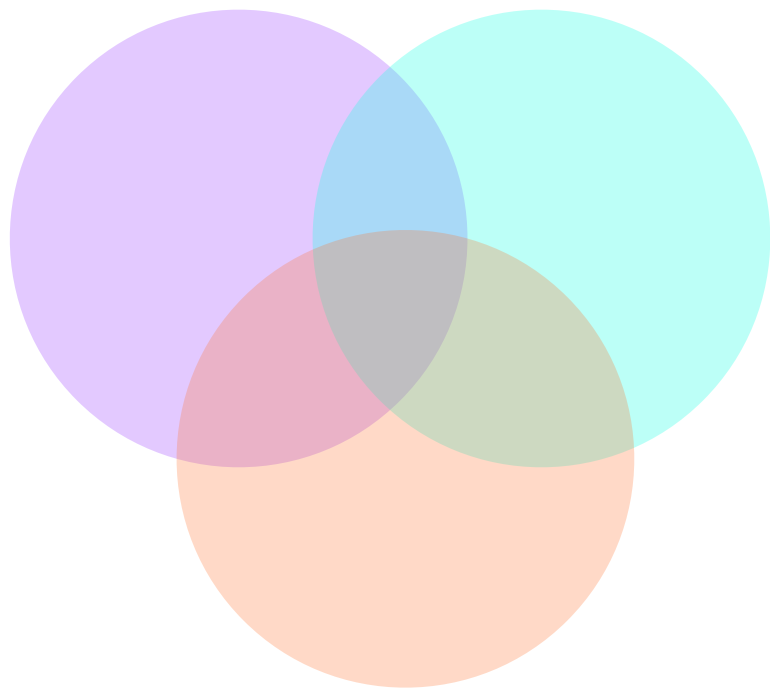




TADs

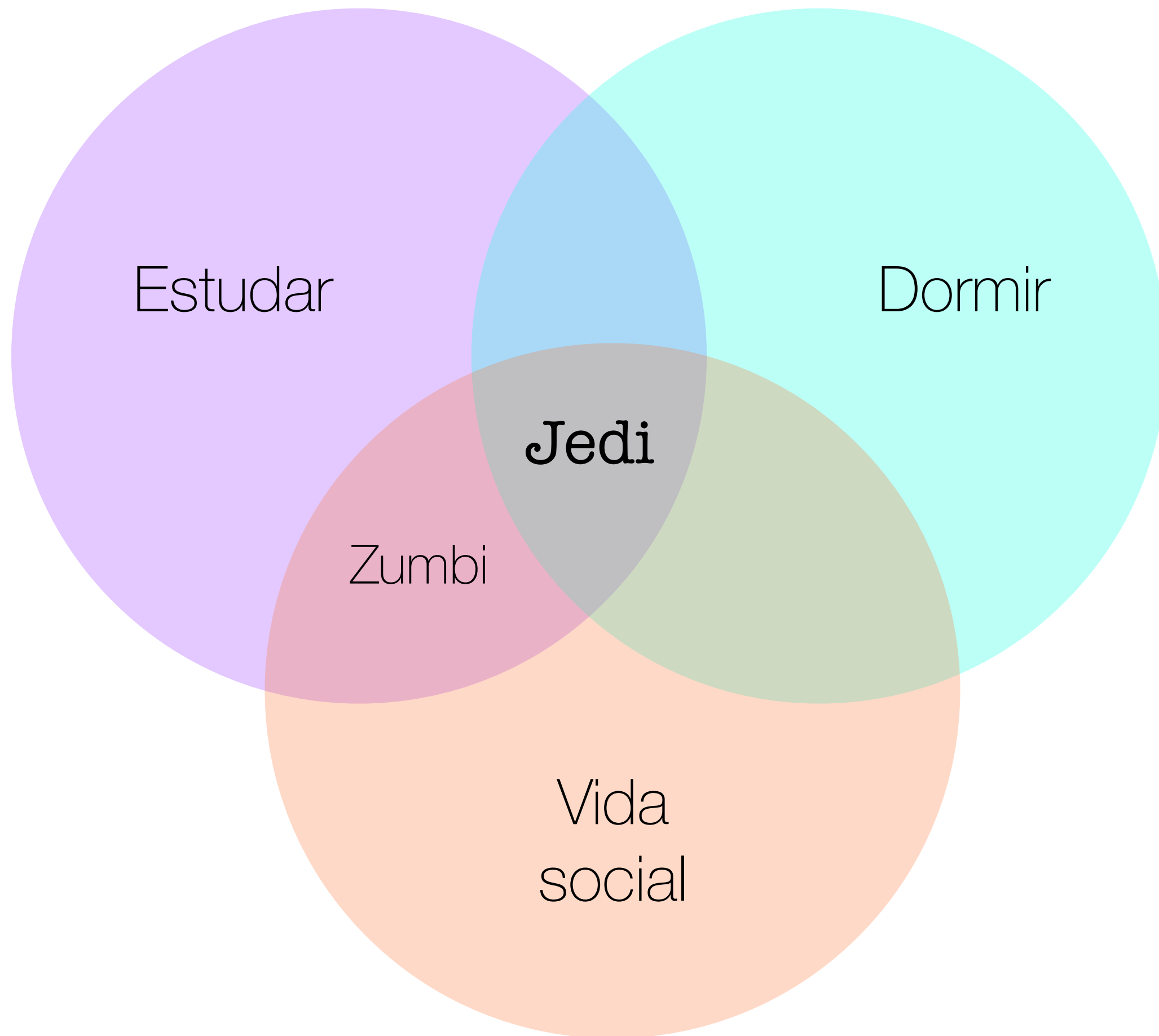
(não-relacionais)





TAD

(conjunto)





TAD

(dicionário)



**mag-ni-fier** (mag'ni-fī-er) *v.*  
that magnifies; spec-  
ing  
**mag-ni-ty** (mag'ni-tē) *n.*  
fier < L. *magnificare*, to  
worship < *magnus*, great  
[Rare] to make greater in  
cause to seem greater, more  
gerate [to magnify one's suf-  
really so; increase the appa-  
lenses 4 [Archaic] to glorify  
increasing the apparent size  
magnifying glass a lens or  
apparent size of an object  
**mag-ni-lo-quent** (mag'ni-lō-  
quence < L. *magniloquent*  
< *magnus*, great (see **MAG**)  
or grandiose in speech or  
— **mag-ni-lo-quence** *n.*  
**Mag-ni-to-gorsk** (māg'ni-  
River; pop. 422,000  
**mag-ni-tude** (mag'ni-  
great; see **MAG**  
importance  
able quan-



conjunto | dicionário  
(não-relacional)

lista | sequência  
pilha | deque | fila  
(linear)

Lista  
Circular

Lista  
Encadeada

Tabela de  
Dispersão

Lista  
Encadeada

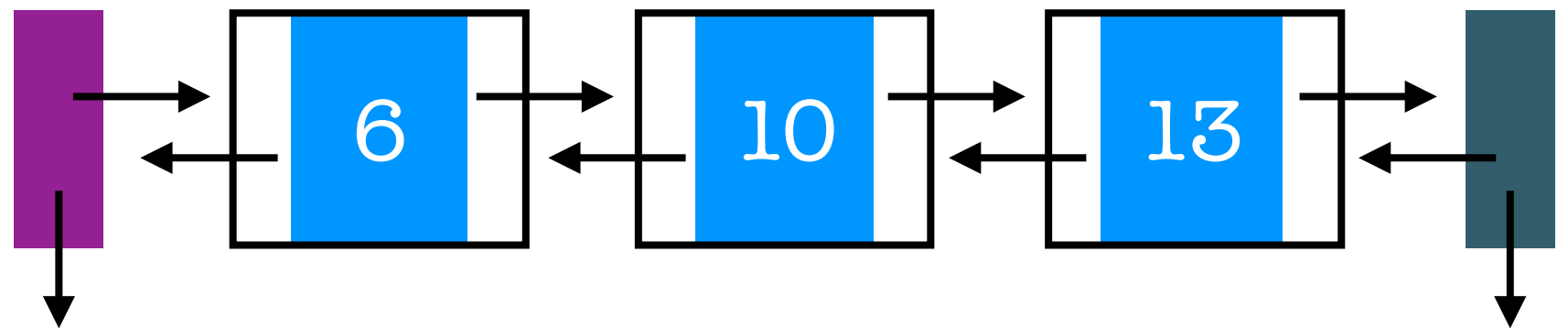
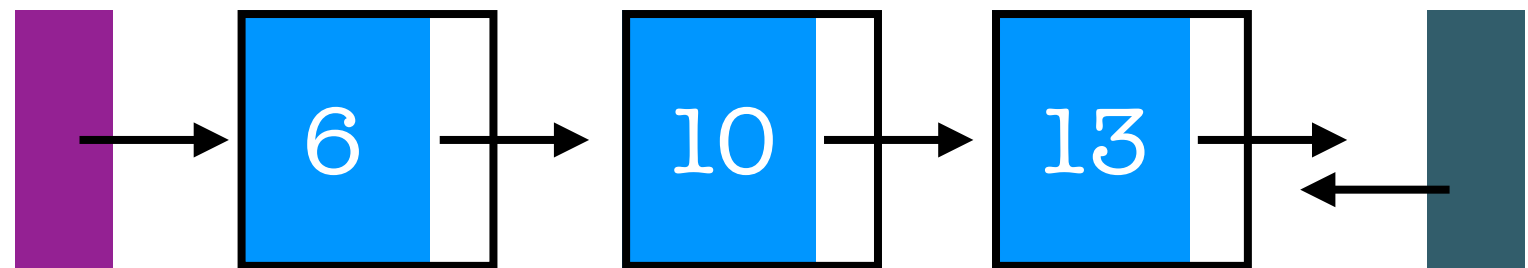
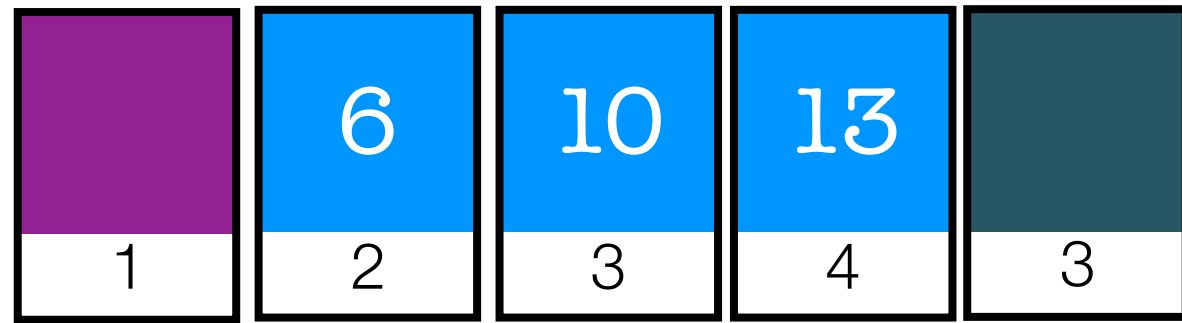
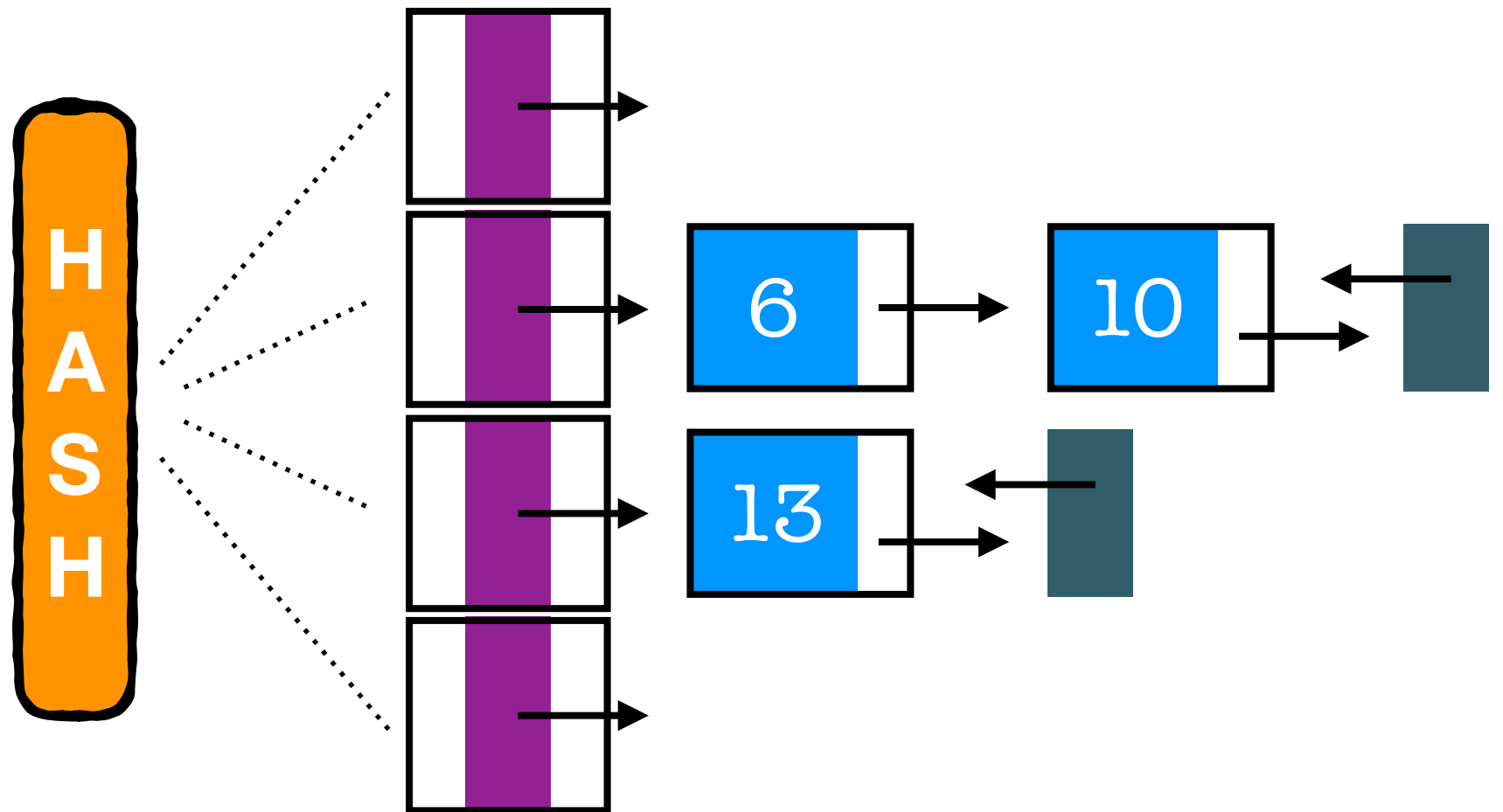
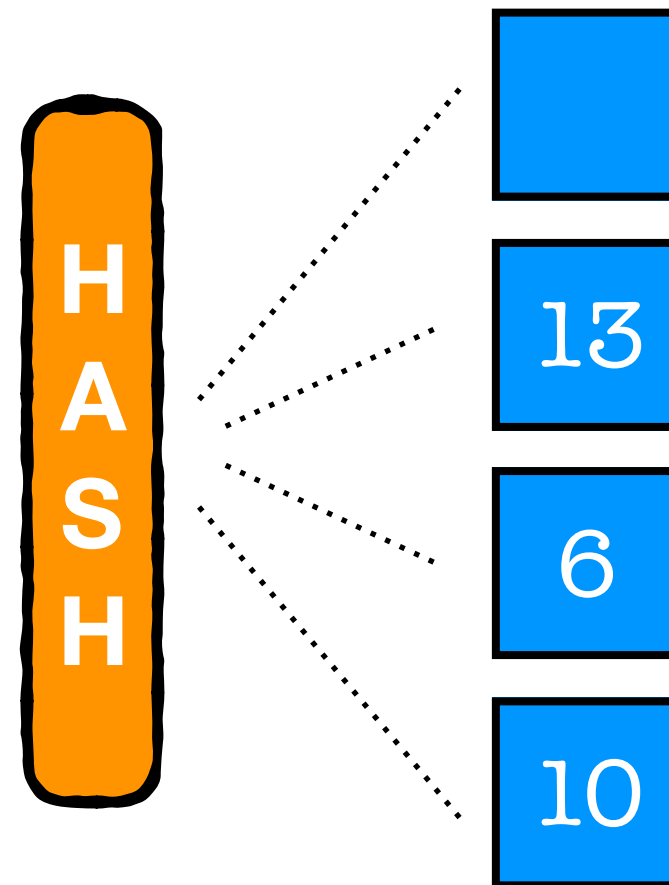


Tabela de  
Dispersão

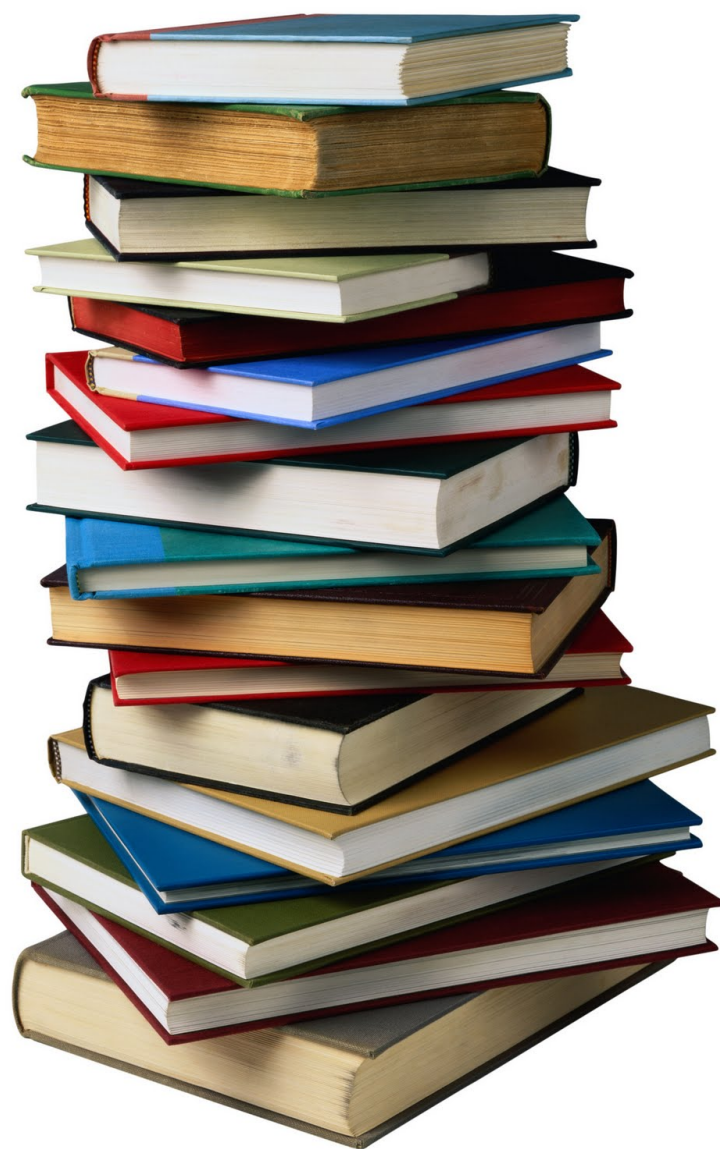


# Tabela de Dispersão



Endereçamento  
**Aberto**





# Ordenação

(algoritmos iterativos)











- Vetor





- Vetor
- Tamanho



- Vetor
- Tamanho
- Tipo



- Vetor
- Tamanho
- Tipo

Comparação





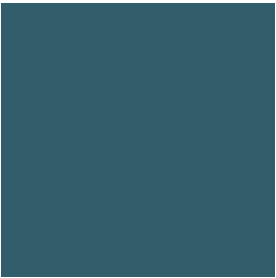
- Vetor
- Tamanho
- Tipo

Comparação



- Vetor
- Tamanho
- Tipo

Comparação









Força bruta  
(bolha)



for each  $i, j$

Força bruta  
(bolha)



for each  $i, j$   
if  $v[i] > v[j]$

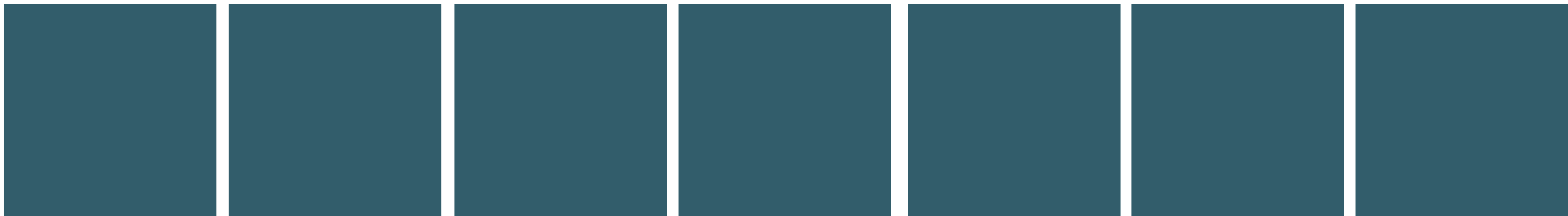
Força bruta  
(bolha)

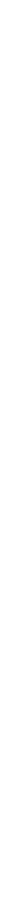




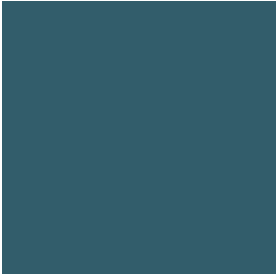
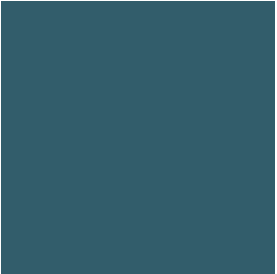
```
for each i, j  
  if  $v[i] > v[j]$   
    swap( $v[i], v[j]$ )
```

Força bruta  
(bolha)





10







10

13







10

13

6



10

13

6

5

7

12

20

11



10

13

6

5

7

12

11

20





**for**  $i = 0, \dots, n-1$

---



```
for i = 0, ..., n-1  
  idx = max(v, n-i)
```



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```





```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



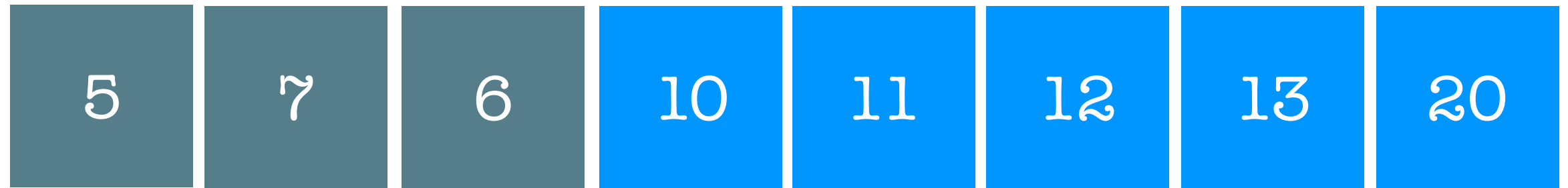
```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
  idx = max(v, n-i)  
  swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)





```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



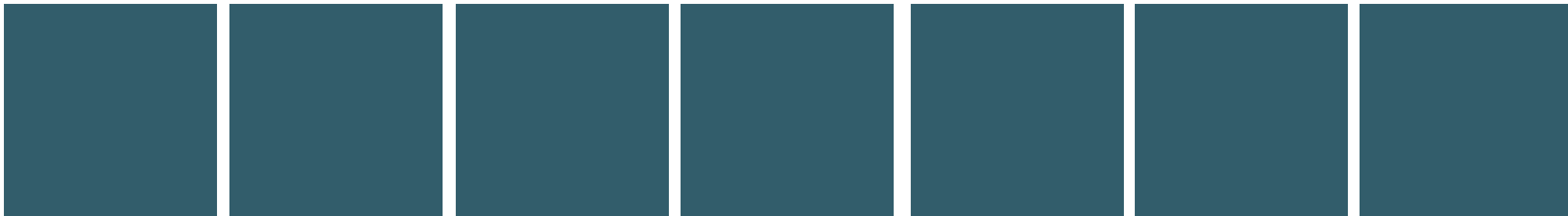
```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)



```
for i = 0, ..., n-1  
    idx = max(v, n-i)  
    swap(v[n-i-1], v[idx])
```

(seleção)





10









10

13











(inserção)



**for**  $i = 1, \dots, n$

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



```
for i = 1, ..., n  
  insert(v, i)
```

(inserção)





**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)





**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



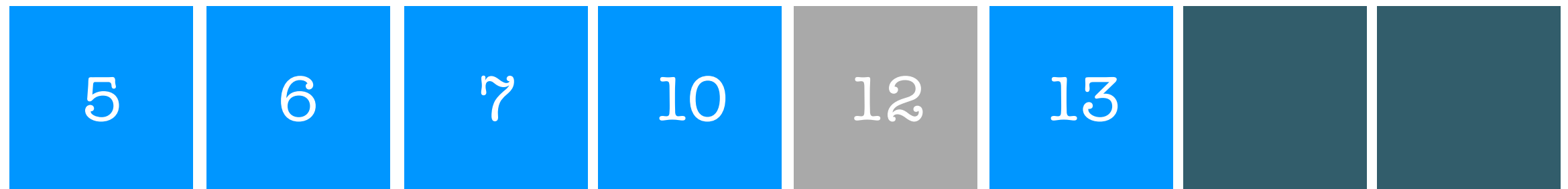
**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



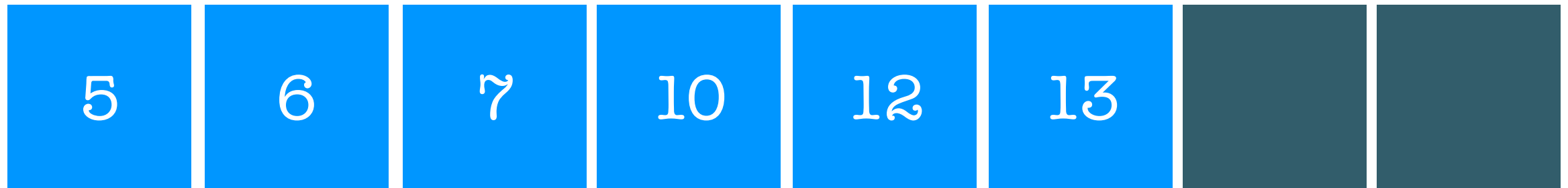
**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)





**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



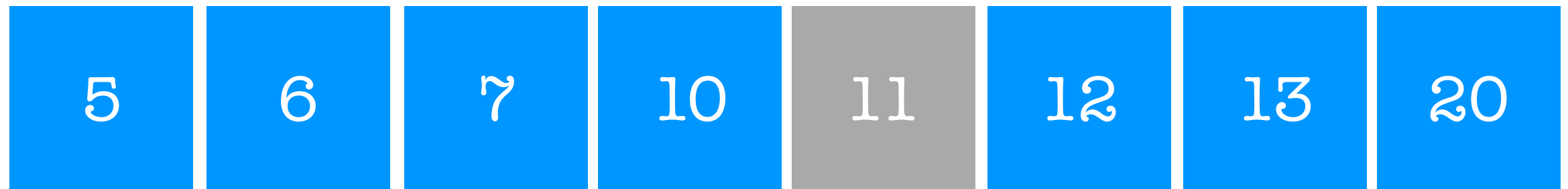
**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
**insert**( $v, i$ )

(inserção)

5	6	7	10	11	12	13	20
---	---	---	----	----	----	----	----

**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)

# Complexidade de algoritmos

Veja slides 02 e 03  
antes de prosseguir



Listas encadeadas





(inserção)



**for**  $i = 1, \dots, n$

(inserção)



```
for i = 1, ..., n  
  insert(v, i)
```

(inserção)



```
for i = 1, ..., n  
  insert(v, i)
```

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)





**for**  $i = 1, \dots, n$   
    **insert**( $v, i$ )

(inserção)



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$

(inserção)



**for**  $i = 1, \dots, n$

$idx = \text{find}(v, i)$

**insert** $(v, i, idx)$

(inserção)



```
for i = 1, ..., n  
  idx = find(v, i)  
  insert(v, i, idx)
```

(inserção)



```
for i = 1, ..., n  
  idx = find(v, i)  
  insert(v, i, idx)
```

(inserção)



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$

(inserção)



```
for i = 1, ..., n  
  idx = find(v, i)  
  insert(v, i, idx)
```

(inserção)



**for**  $i = 1, \dots, n$

$idx = \text{find}(v, i)$

**insert** $(v, i, idx)$

(inserção)





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$

(inserção)



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$

(inserção)



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(n)$





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(n)$



$O(n)$





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(n)$



$O(n)$

$O(n^2)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$



$O(\log n)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$



$O(\log n)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$



$O(\log n)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

**insert** $(v, i, \text{idx})$



$O(\log n)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$   $\leftarrow O(\log n)$

$\text{insert}(v, i, \text{idx})$   $\leftarrow O(n)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(\log n)$



$O(n)$

$O(n^2)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(\log n)$





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

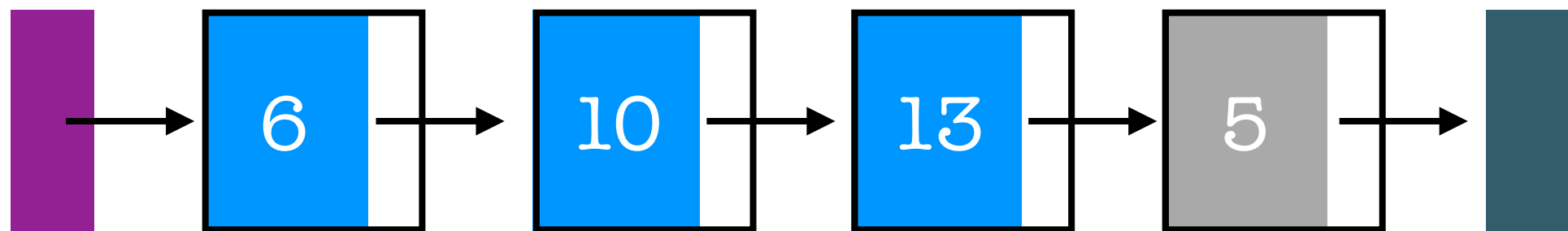
**insert** $(v, i, \text{idx})$

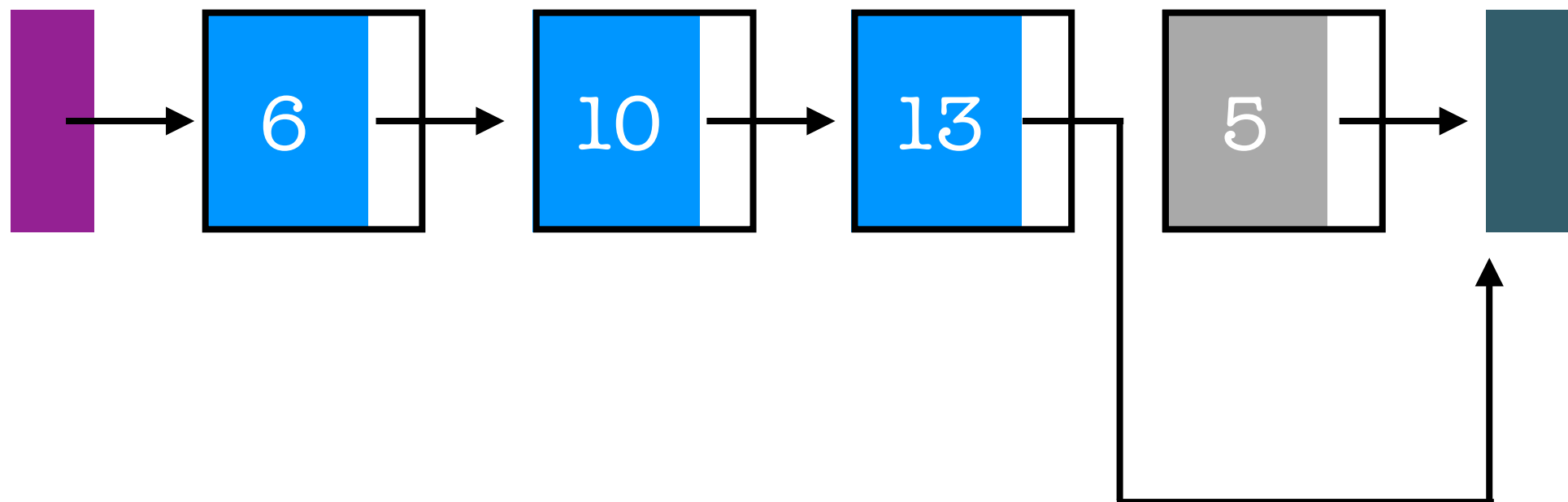


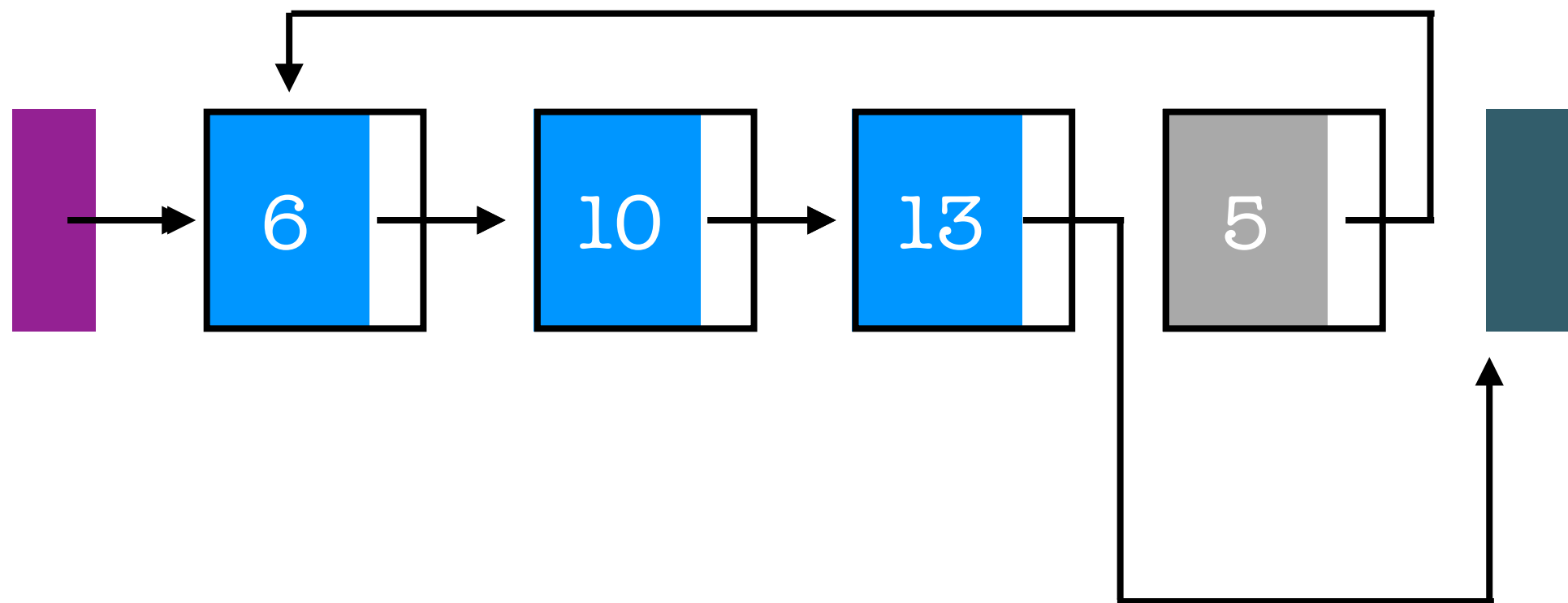
$O(\log n)$

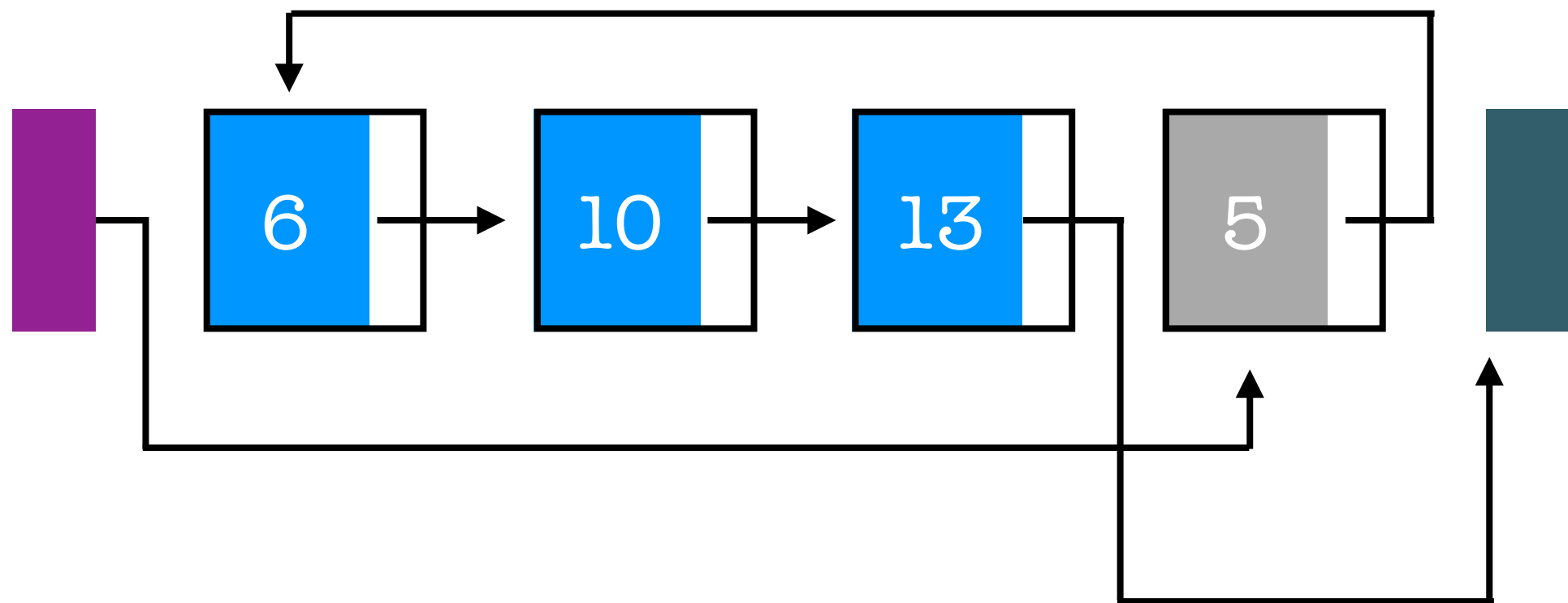


Listas encadeadas



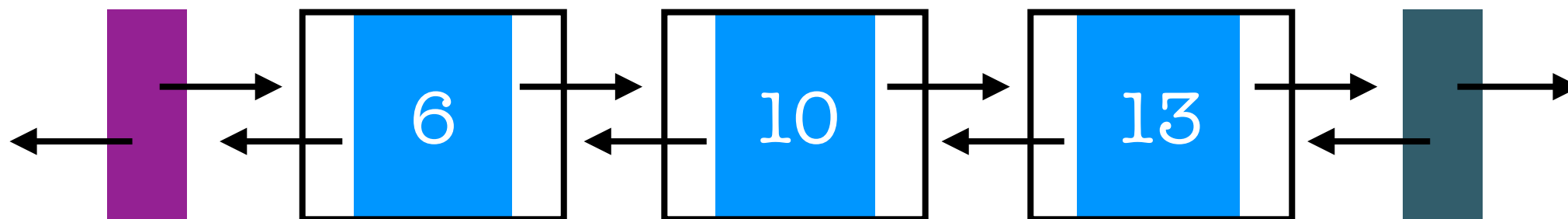




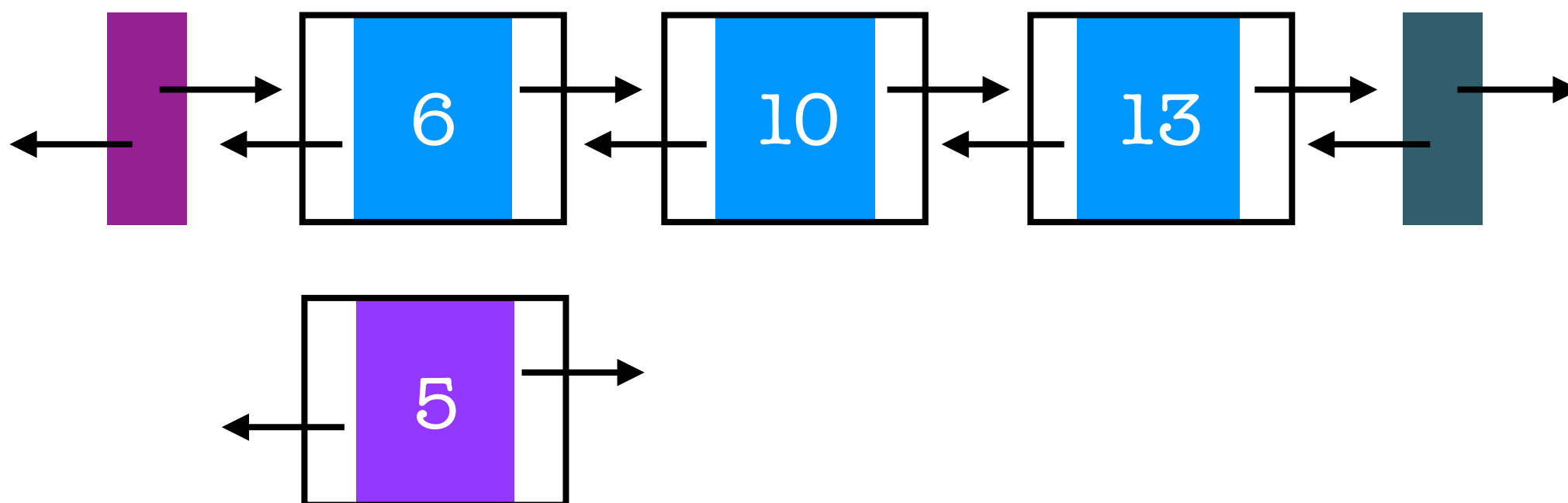


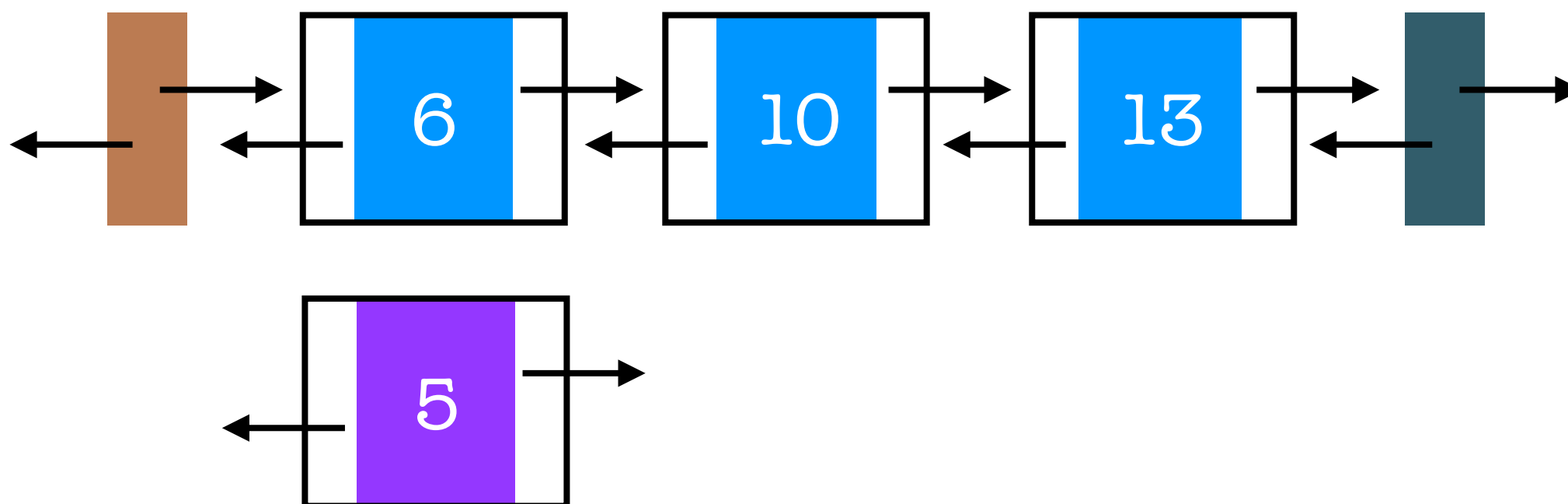


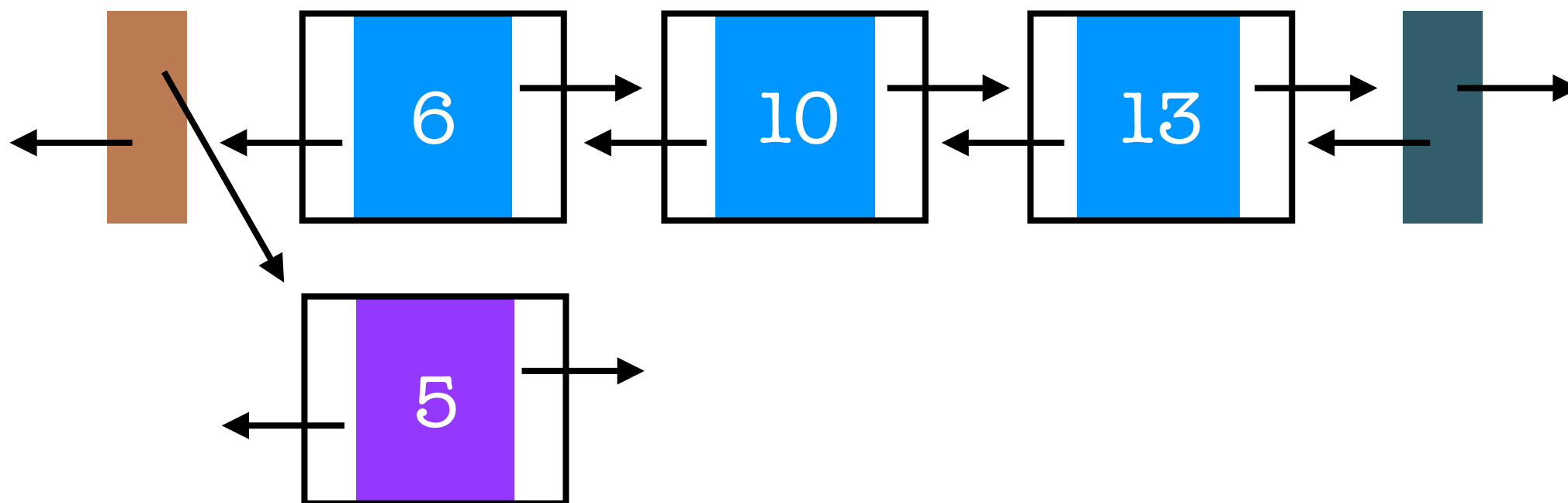
listas **duplamente** encadeadas

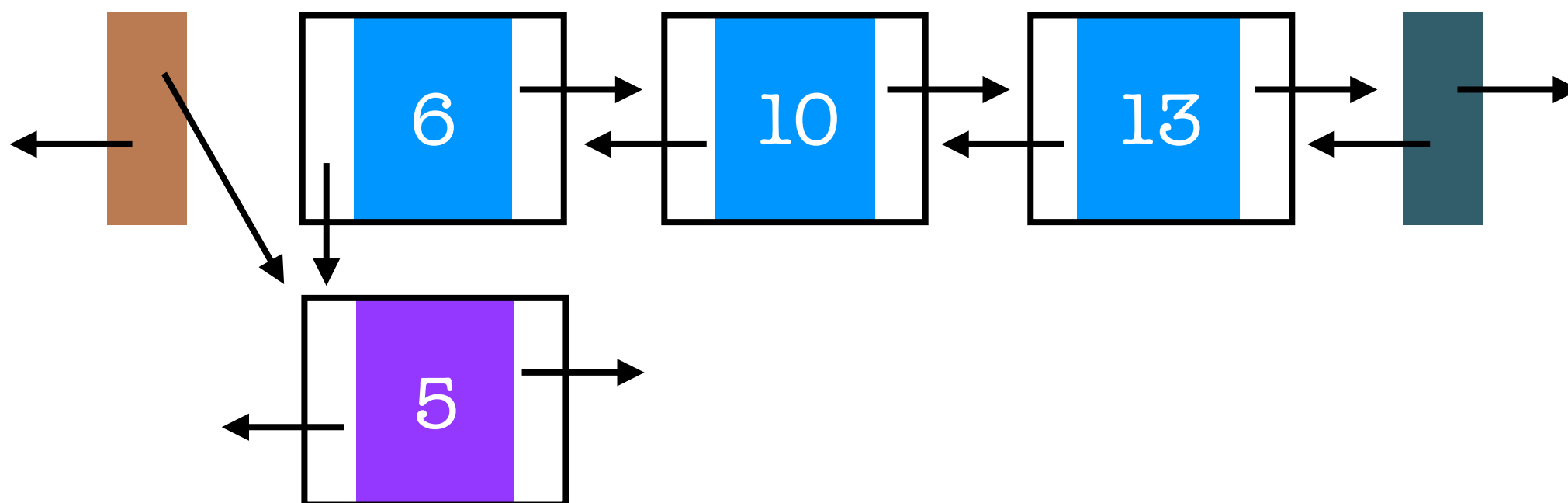


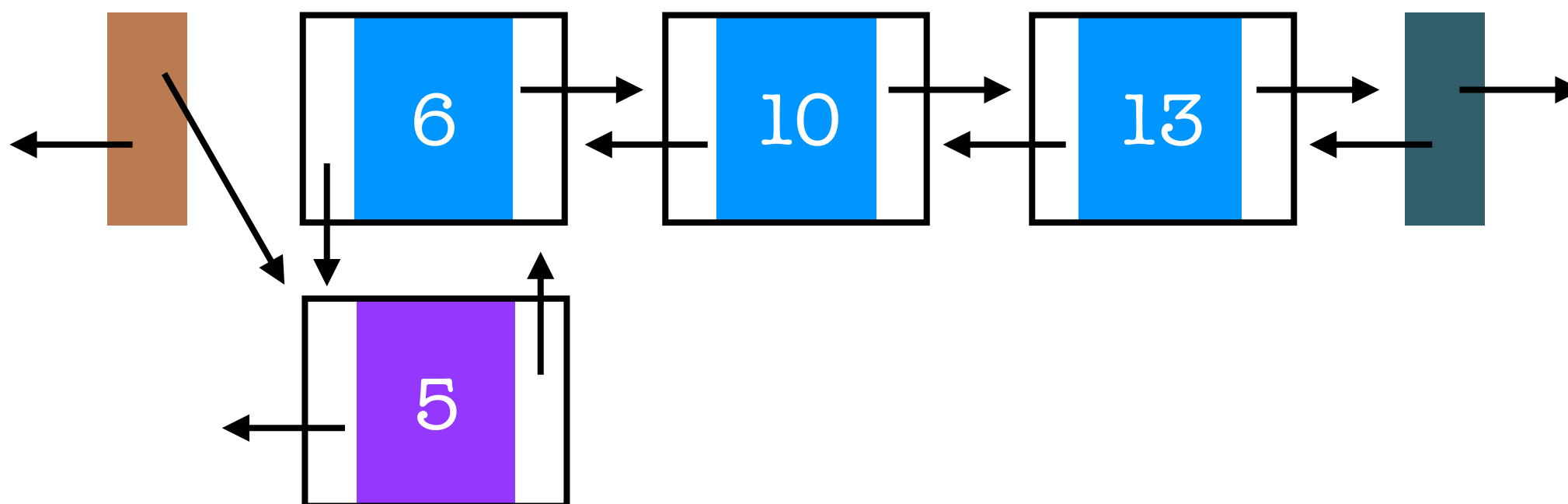


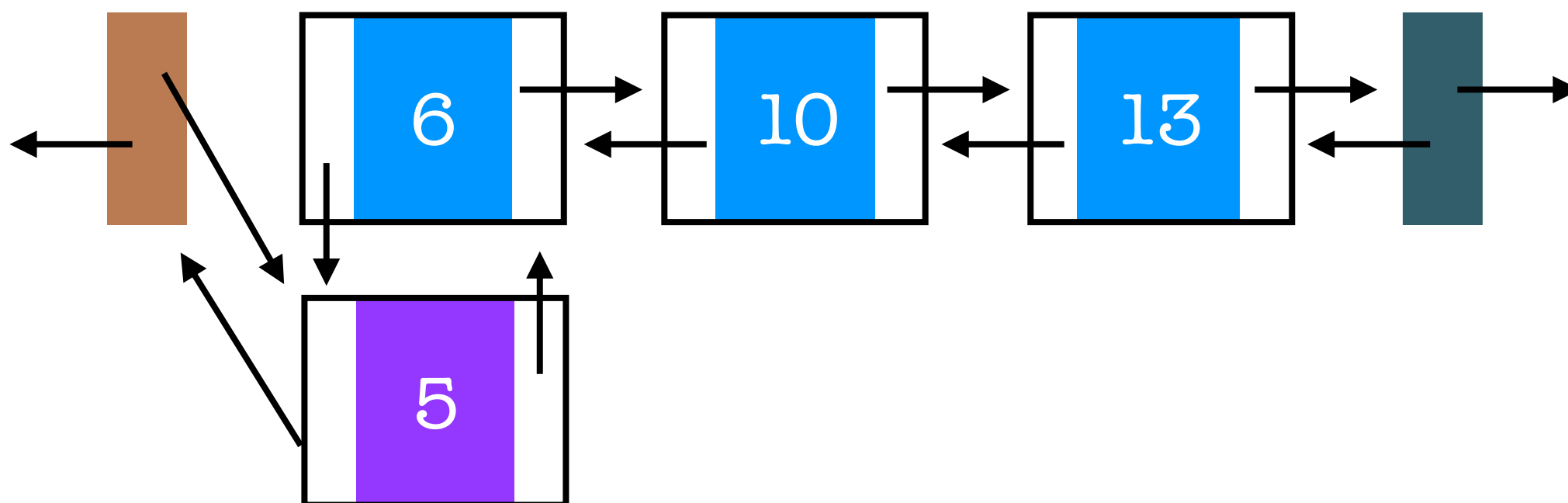














TAD Lista		
Operação	Vetor	Lista Encadeada
acesso		



TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca		

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho		

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$



TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção		

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção	$O(n)$	

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção	$O(n)$	$O(1)$

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção	$O(n)$	$O(1)$
remoção		

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção	$O(n)$	$O(1)$
remoção	$O(n)$	

TAD Lista		
Operação	Vetor	Lista Encadeada
acesso	$O(1)$	$O(n)$
busca	$O(n)$	$O(n)$
tamanho	$O(1)$	$O(1)$
inserção	$O(n)$	$O(1)$
remoção	$O(n)$	$O(1)$



```
for i = 1, ..., n  
  idx = find(v, i)  
  insert(v, i, idx)
```



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$





**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$        $\longleftarrow O(1)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(1)$

$O(n^2)$



**for**  $i = 1, \dots, n$

$\text{idx} = \text{find}(v, i)$

$\text{insert}(v, i, \text{idx})$



$O(n)$



$O(1)$

$O(n^2)$