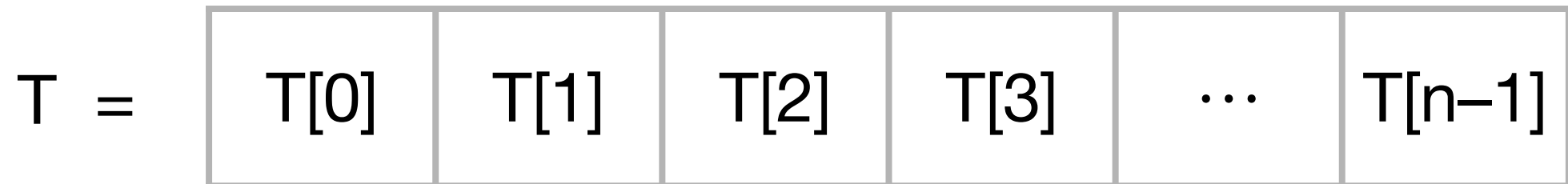


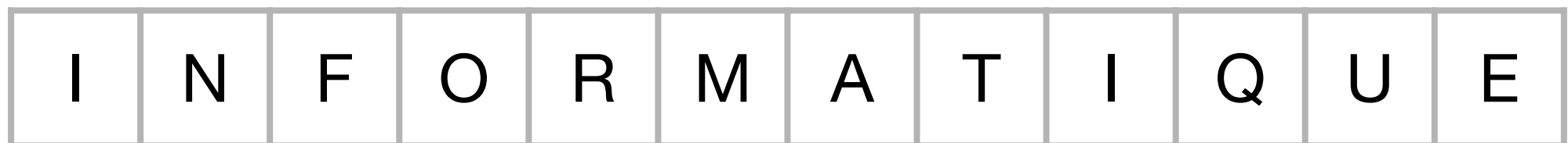
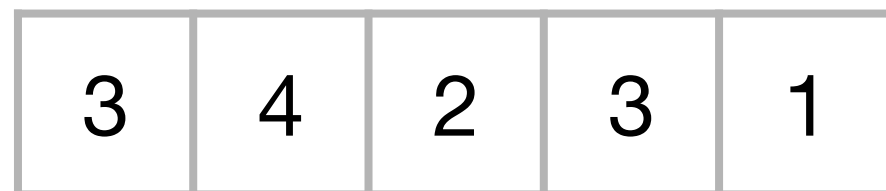
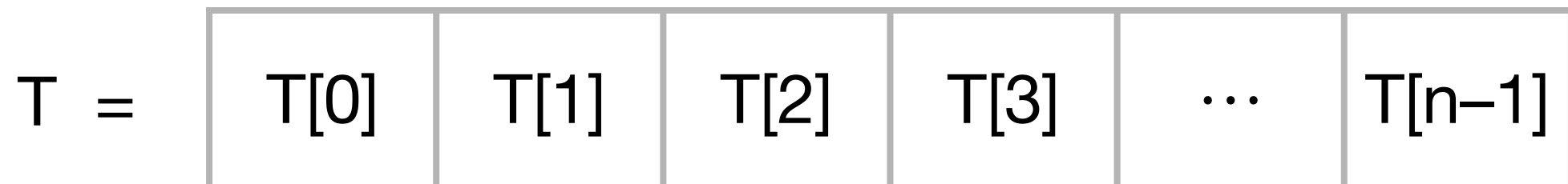
Introduction à l'informatique CM4

Antonio E. Porreca
aeporreca.org/introinfo

Structures de données : les tableaux



Structures de données : les tableaux



Parcourir un tableau

```
procedure parcours(T)
  n := longueur(T)
  i := 0
  tant que i < n faire
    écrire T[i]
    i := i + 1
  fin tant que
fin procedure
```

Avec la boucle « pour »

```
procedure parcours(T)
  n := longueur(T)
  i := 0
  tant que i < n faire
    écrire T[i]
    i := i + 1
  fin tant que
fin procedure
```

=

```
procedure parcours(T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    écrire T[i]
  fin pour
fin procedure
```

Recherche dans un tableau

```
fonction chercher(x, T)
  n := longueur(T)
  i := 0
  tant que i < n faire
    si T[i] = x alors
      retourner i
    fin si
    i := i + 1
  fin tant que
  retourner -1
fin fonction
```

Recherche dans un tableau

```
fonction chercher(x, T)
  n := longueur(T)
  i := 0
  tant que i < n faire
    si T[i] = x alors
      retourner i
    fin si
    i := i + 1
  fin tant que
  retourner -1
fin fonction
```

=

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

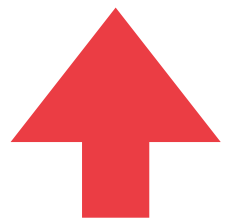
Recherche dans un tableau

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```


Recherche linéaire

Recherche de 33

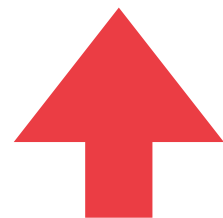
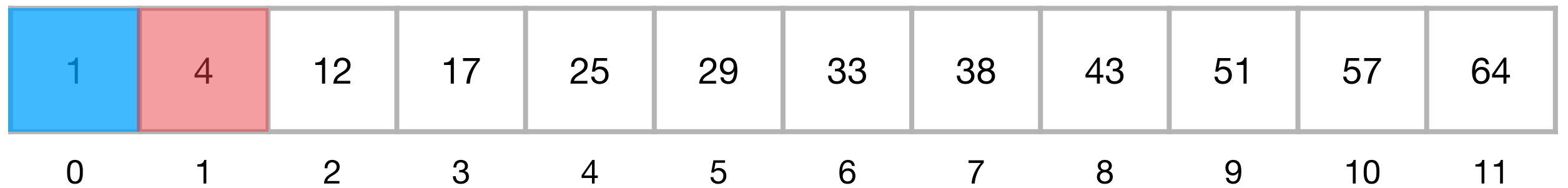
1	4	12	17	25	29	33	38	43	51	57	64
0	1	2	3	4	5	6	7	8	9	10	11



i

Recherche linéaire

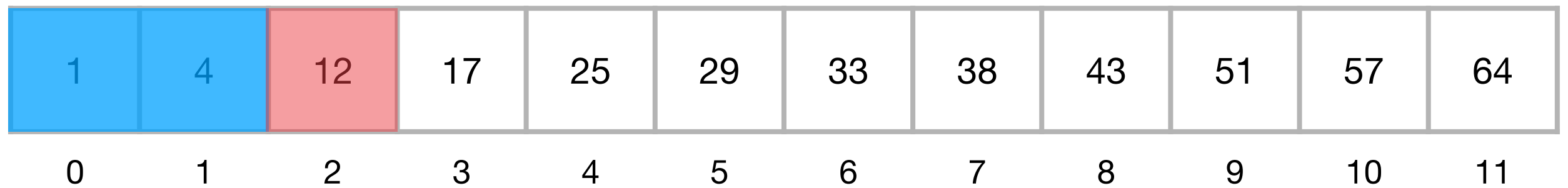
Recherche de 33



i

Recherche linéaire

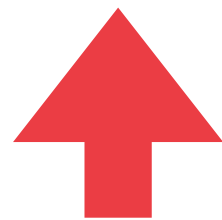
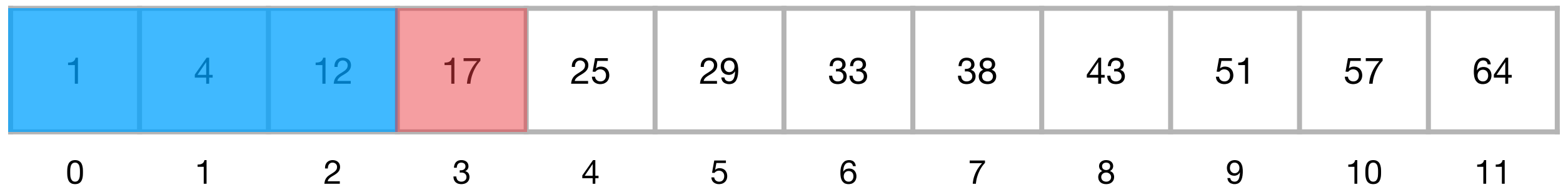
Recherche de 33



i

Recherche linéaire

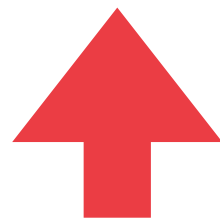
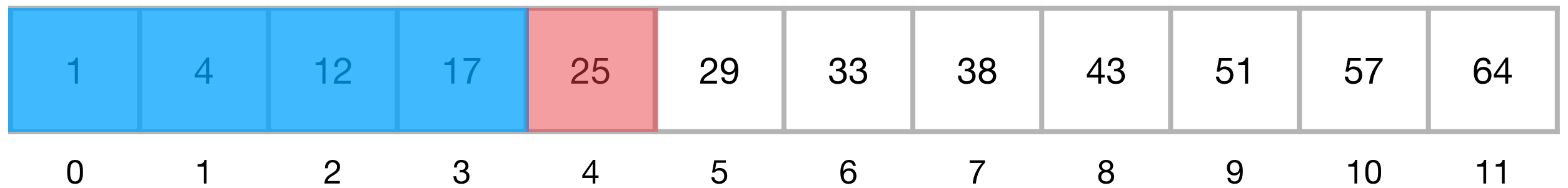
Recherche de 33



i

Recherche linéaire

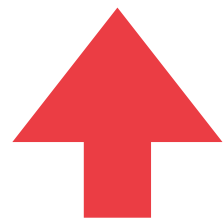
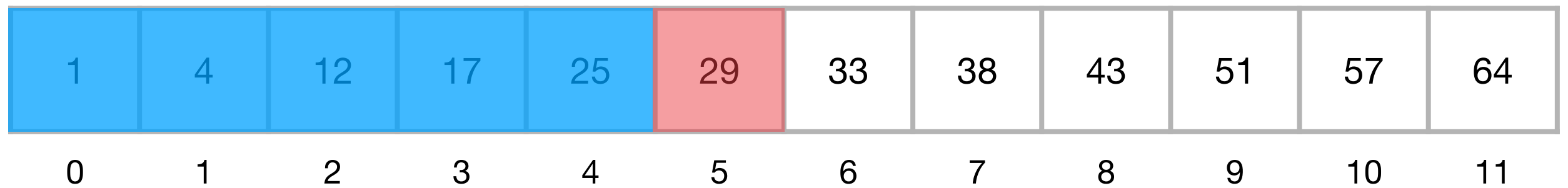
Recherche de 33



i

Recherche linéaire

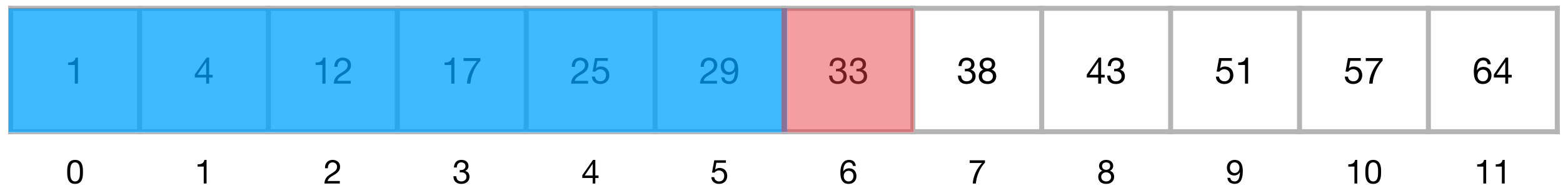
Recherche de 33



i

Recherche linéaire

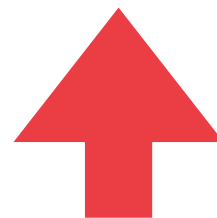
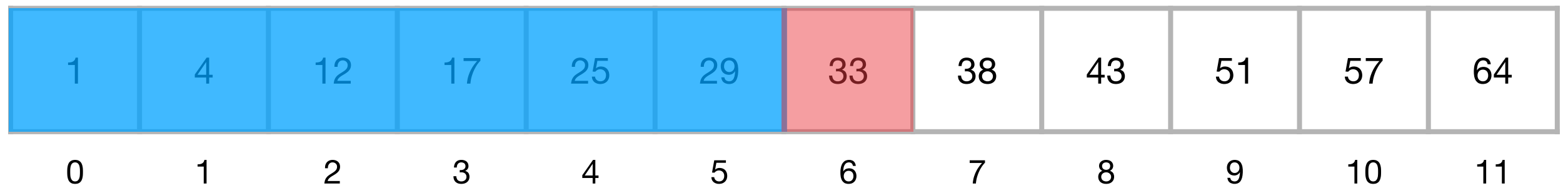
Recherche de 33



i

Recherche linéaire

Recherche de 33

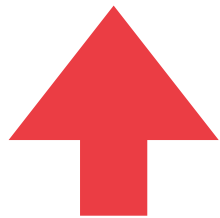


i

Recherche linéaire

Recherche de 3

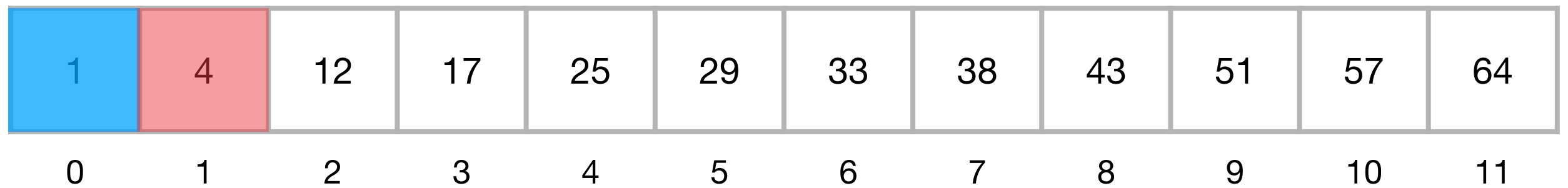
1	4	12	17	25	29	33	38	43	51	57	64
0	1	2	3	4	5	6	7	8	9	10	11



i

Recherche linéaire

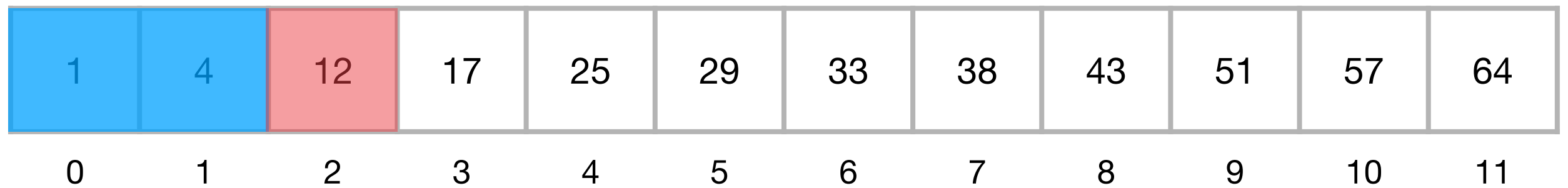
Recherche de 3



i

Recherche linéaire

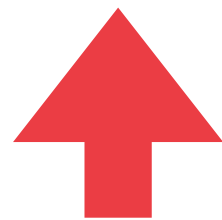
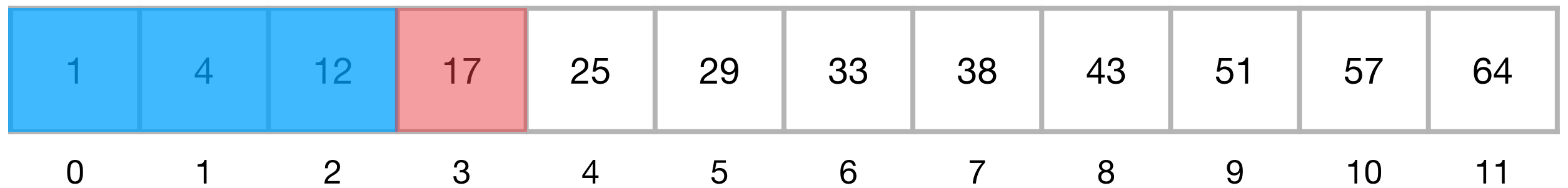
Recherche de 3



i

Recherche linéaire

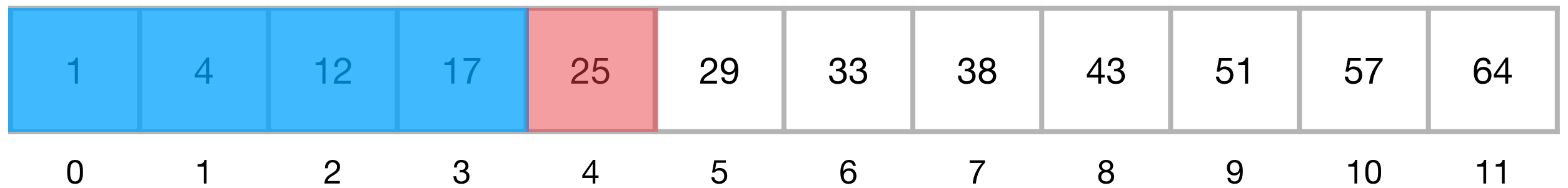
Recherche de 3



i

Recherche linéaire

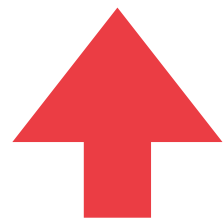
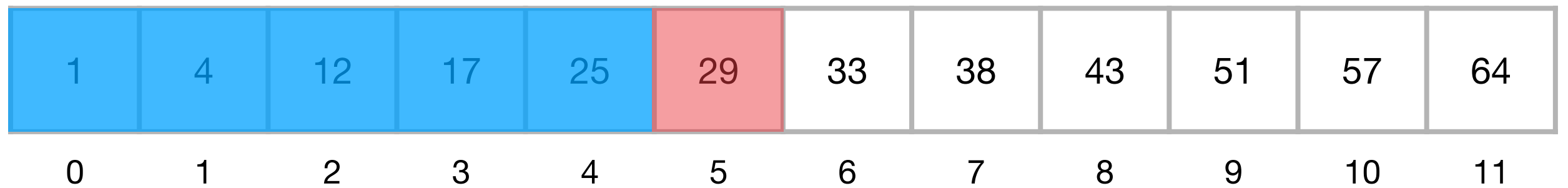
Recherche de 3



i

Recherche linéaire

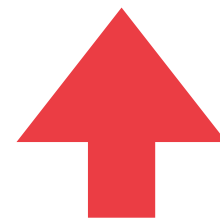
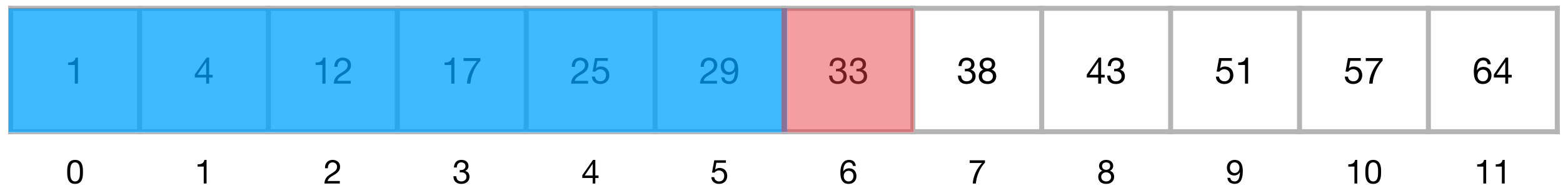
Recherche de 3



i

Recherche linéaire

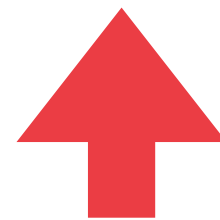
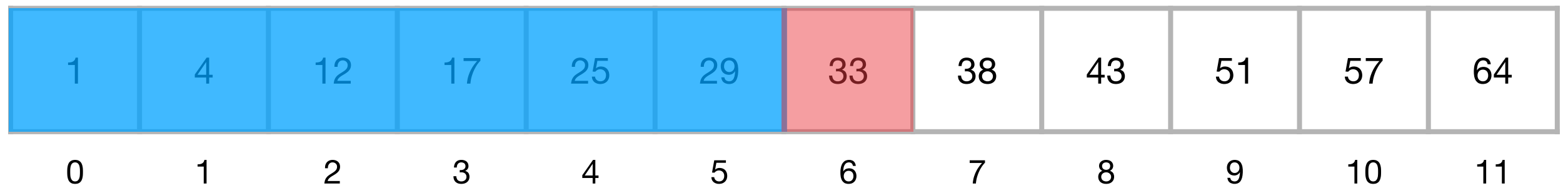
Recherche de 3



i

Recherche linéaire

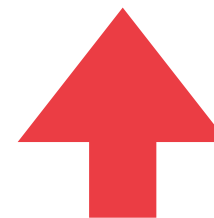
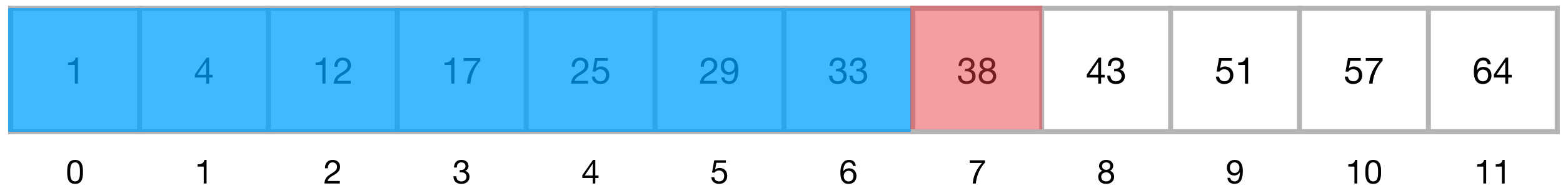
Recherche de 3



i

Recherche linéaire

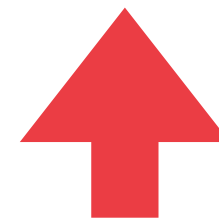
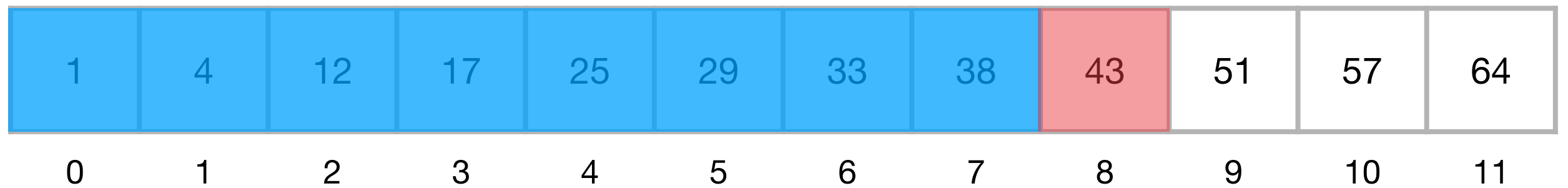
Recherche de 3



i

Recherche linéaire

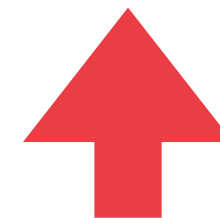
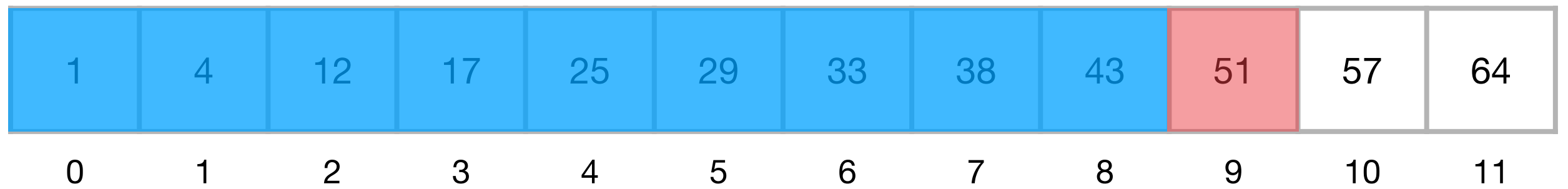
Recherche de 3



i

Recherche linéaire

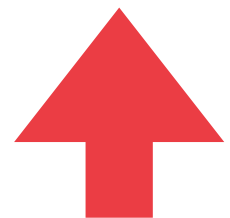
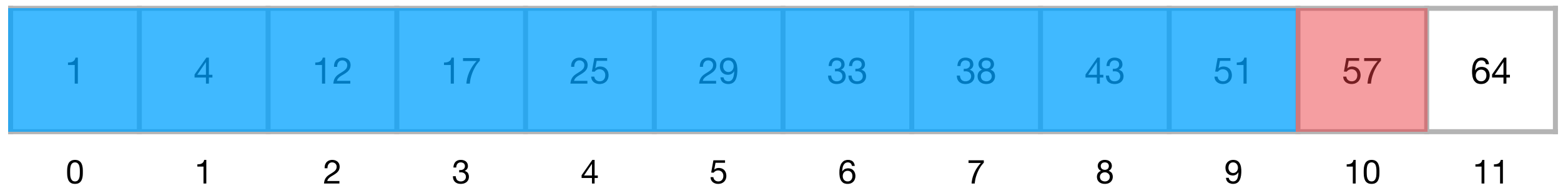
Recherche de 3



i

Recherche linéaire

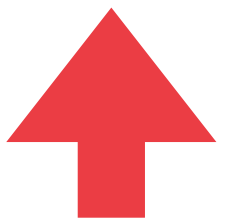
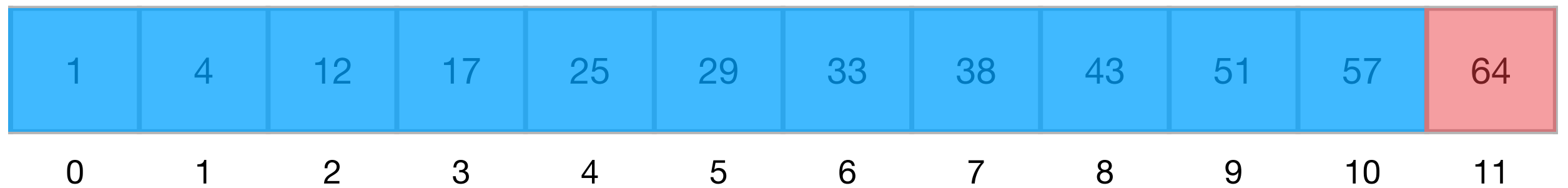
Recherche de 3



i

Recherche linéaire

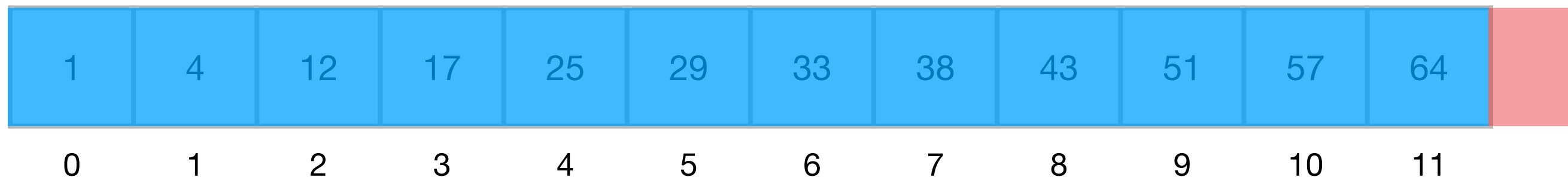
Recherche de 3



i

Recherche linéaire

Recherche de 3



i

Recherche linéaire

Recherche de 3

1	4	12	17	25	29	33	38	43	51	57	64
0	1	2	3	4	5	6	7	8	9	10	11



i

Recherche dans un tableau

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

Terminaison ?

Correction ?

Efficacité ?

Terminaison

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

- Au début, on a $i = 0$
- Il reste toujours $n - i$ positions à examiner
- i est incrémenté à chaque itération
- Tôt ou tard soit on trouve x , soit on arrive à $i = n$, et l'algorithme termine

Correction

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

- **Invariant de boucle** : si x est dans le tableau, alors il se trouve dans le sous-tableau $T[i, \dots, n - 1]$
 - C'est vrai au début de l'algorithme
 - Ça reste vrai à chaque itération de la boucle, parce qu'on vérifie toujours si $T[i] = x$
- Si on sort de la boucle avec $i = n$, alors si x est dans le tableau, il est dans le sous-tableau vide $T[n, n - 1]$, c'est à dire qu'il n'est pas là

Comptage des opérations

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

Comptage des opérations

#opérations

👉 **fonction** chercher(x, T)
 n := longueur(T)
 pour i := 0 à n - 1 **faire**
 si T[i] = x **alors**
 retourner i
 fin si
 fin pour
 retourner -1
fin fonction

Comptage des opérations

#opérations



```
fonction chercher(x, T)  
  n := longueur(T)  
  pour i := 0 à n - 1 faire  
    si T[i] = x alors  
      retourner i  
    fin si  
  fin pour  
  retourner -1  
fin fonction
```

1

Comptage des opérations

#opérations

fonction chercher(x, T)

n := longueur(T)

1



pour i := 0 à n - 1 **faire**

1

si T[i] = x **alors**

retourner i

fin si

fin pour

retourner -1

fin fonction

Comptage des opérations

#opérations

fonction chercher(x, T)

n := longueur(T)

1

pour i := 0 à n - 1 **faire**

1



si T[i] = x **alors**

1

retourner i

fin si

fin pour

retourner -1

fin fonction

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire  1
    si T[i] = x alors        1
      retourner i           1
    fin si
  fin pour
  retourner -1
fin fonction
```



Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire   1
    si T[i] = x alors         1
      retourner i             1
    fin si
  fin pour
  retourner -1
fin fonction
```



Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire  1
    si T[i] = x alors        1
      retourner i           1
    fin si
  fin pour
  retourner -1
fin fonction
```



Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire   1
    si T[i] = x alors        1
      retourner i           1
    fin si
  fin pour
  retourner -1               1
fin fonction
```



Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire  1
    si T[i] = x alors        1
      retourner i           1
    fin si
  fin pour
  retourner -1               1
👉 fin fonction
```

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire   1
    si T[i] = x alors         1
      retourner i             1
    fin si
  fin pour
  retourner -1                1
fin fonction
```

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire   1
    si T[i] = x alors         1
      retourner i             1
    fin si
  fin pour
  retourner -1                1
fin fonction
```

si on a $T[k] = x$

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire  1
    si T[i] = x alors        1
      retourner i           1
    fin si
  fin pour
  retourner -1               1
fin fonction
```

} k + 1 fois

si on a $T[k] = x$

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

1

1

1

1

1

}

k + 1 fois

si on a $T[k] = x$

$2(k + 1) + 2$
opérations

Comptage des opérations

#opérations

```
fonction chercher(x, T)
  n := longueur(T)           1
  pour i := 0 à n - 1 faire  1      n+1 fois
    si T[i] = x alors        1      n fois
      retourner i           1
    fin si
  fin pour
  retourner -1               1
fin fonction
```

si x n'est pas là

Comptage des opérations

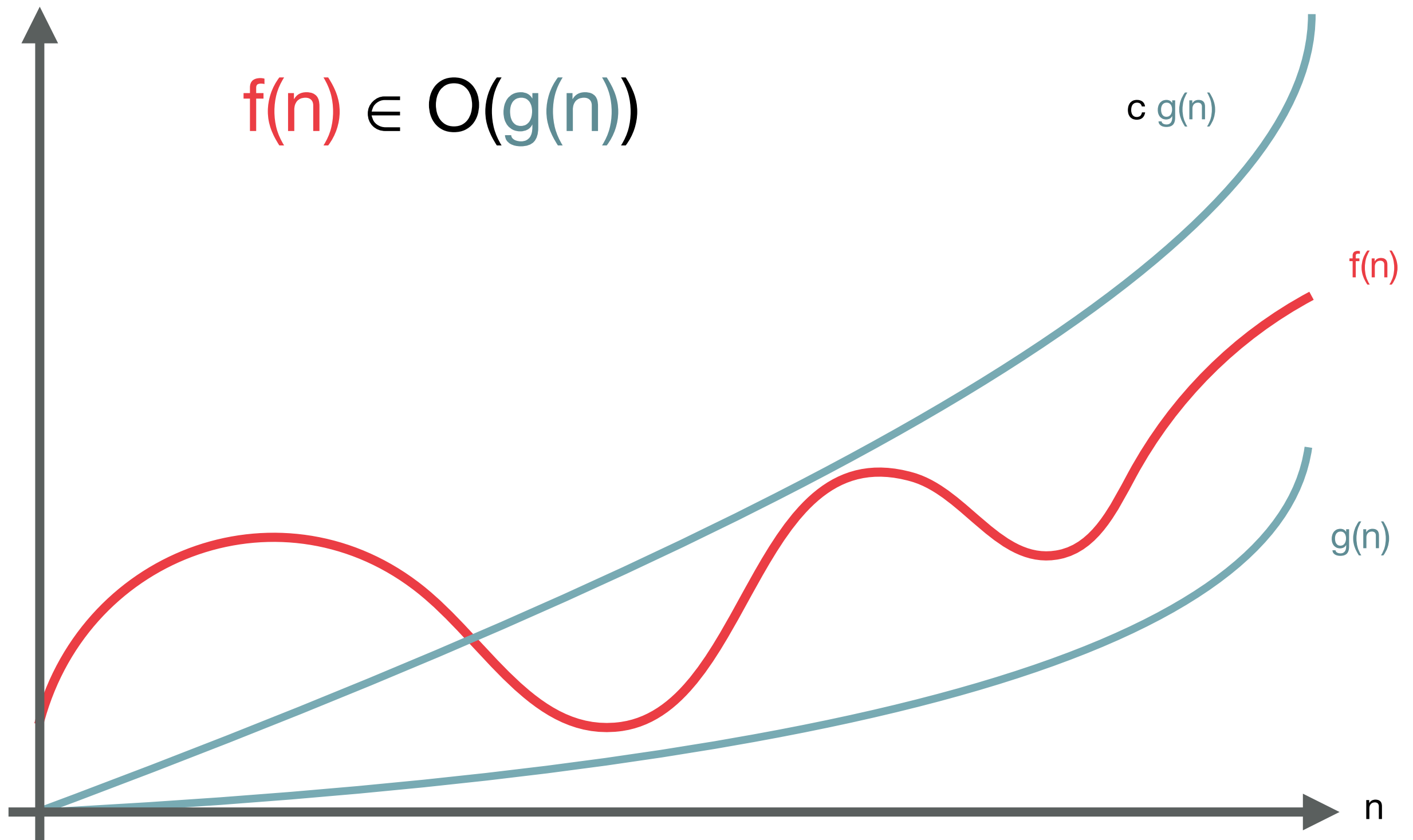
	#opérations	
fonction chercher(x, T)		
n := longueur(T)	1	
pour i := 0 à n - 1 faire	1	n+1 fois
si T[i] = x alors	1	n fois
retourner i	1	
fin si		
fin pour		
retourner -1	1	
fin fonction		
		2n + 3
si x n'est pas là		opérations

Efficacité

```
fonction chercher(x, T)
  n := longueur(T)
  pour i := 0 à n - 1 faire
    si T[i] = x alors
      retourner i
    fin si
  fin pour
  retourner -1
fin fonction
```

- Si on a de la chance, on a $T[0] = x$ et on termine tout de suite en **4 opérations**
- Si $T[k] = x$ on fait $2(k + 1) + 2 = \mathbf{2k + 4}$ **opérations**
- Si x n'est pas là on fait **$2n + 3$ opérations**

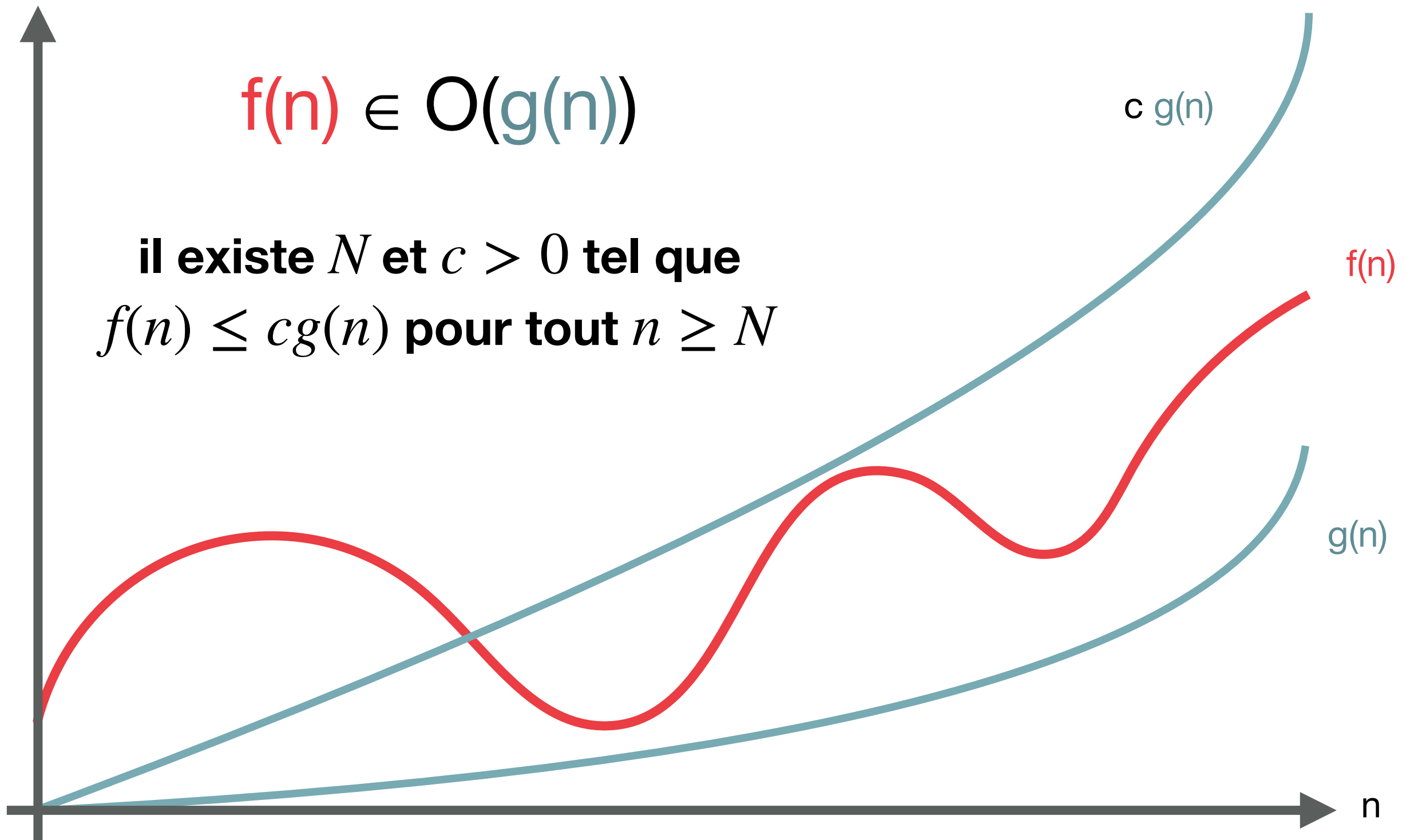
Notation « grand O »



Notation « grand O »

$$f(n) \in O(g(n))$$

il existe N et $c > 0$ tel que
 $f(n) \leq cg(n)$ pour tout $n \geq N$



Ordres de grandeur

Il existe N et $c > 0$ tel que $f(n) \leq cg(n)$ pour tout $n \geq N$

- $n \in O(n)$
- $n + 5 \in O(n)$
- $2n + 5 \in O(n)$
- $n^2 + 2 \in O(n^2)$
- $n^2 \notin O(n)$

**Recherche dans
un annuaire
ou un dictionnaire ?**