

# Partie 2 : Types structurés-Tableaux

# Rappel : Types de base

- ▶ Entiers (int)
- ▶ Réels (float)
- ▶ Booléens (True/False)
- ▶ Caractère
- ▶ Pointeur (adresse)
- ▶ ...

# Tableaux

## Définition :

Un tableau à une dimension

`<type> <identificateur> [N]`

est :

- ▶ une structure de données qui contient **N variables de même type**
- ▶ chacune accessible par une clé (ou indice) comprise entre 0 et N-1
- ▶ le nombre d'éléments du tableau (**dimension ou taille**) est constant.

# Tableaux

## Définition :

- ▶ l'accès aux éléments s'effectue directement par la clé
- ▶ pour accéder à l'élément d'indice  $i$  on écrit  $T[i]$ .
- ▶ Le **temps de calcul** pour **accéder à un élément du tableau** est **constant**.

## Tableaux : exemples

**T**[0...9] de car #tableau T de 10 caracteres.

**T**[0] = 'a' #affectation de la 1<sup>ère</sup> case de T.

	0	1		i					9	
T	a	g	h	d	b	t	v	y	q	u

Que vaut T[3] ?

# Tableaux : exemples

Initialiser le tableau  $T$  pour que  $T[i]$  contienne  $x$  :

```
T[0...9] tableau de 10 entiers
```

```
for i in range(10):
```

```
    T[i] = x
```

Initialiser le tableau  $T$  pour que  $T[i]$  contienne  $i*i$  :

```
T[0...9] tableau de 10 entiers
```

```
for i in range(10):
```

```
    T[i] = i*i
```

# Tableaux

## Définition :

Un tableau à k dimensions

`<type> <identificateur> [N1] [N2]...[Nk]`

est :

- ▶ une structure de données qui contient  $N_1 * N_2 * \dots * N_k$   
**variables de même type**
- ▶ Les indices de la j<sup>ème</sup> dimension varient entre 0 et  $N_j - 1$
- ▶ le nombre d'éléments du tableau (**dimension ou taille**) est constant.

# Tableaux

## Définition :

- ▶ l'accès aux éléments s'effectue directement par les clés

Par exemple :

si  $T$  est un tableau à 2 dimensions pour accéder à l'élément d'indice  $(i, j)$  on écrit  $T[i][j]$ .

- ▶ Le **temps de calcul** pour **accéder à un élément du tableau** est **constant**.

# Tableaux : exemples

**mat**[0...4][0...2] tableau de 5\*3 entiers

*# mat est une matrice d'entiers avec 5 lignes et 3 colonnes.*

**mat**[4][2] = 7

*#affectation de la "derniere" case de mat.*

		0	1	2
0				
1				
2				
3				
4				7

# Tableaux : exemples

Initialiser la matrice 5 x 3 pour que `mat[i][j]` contienne 0

```
mat[0..4][0..2] tableau de 5*3 entiers
```

```
for i in range(5):  
  for j in range(3):  
    mat[i][j] = 0
```

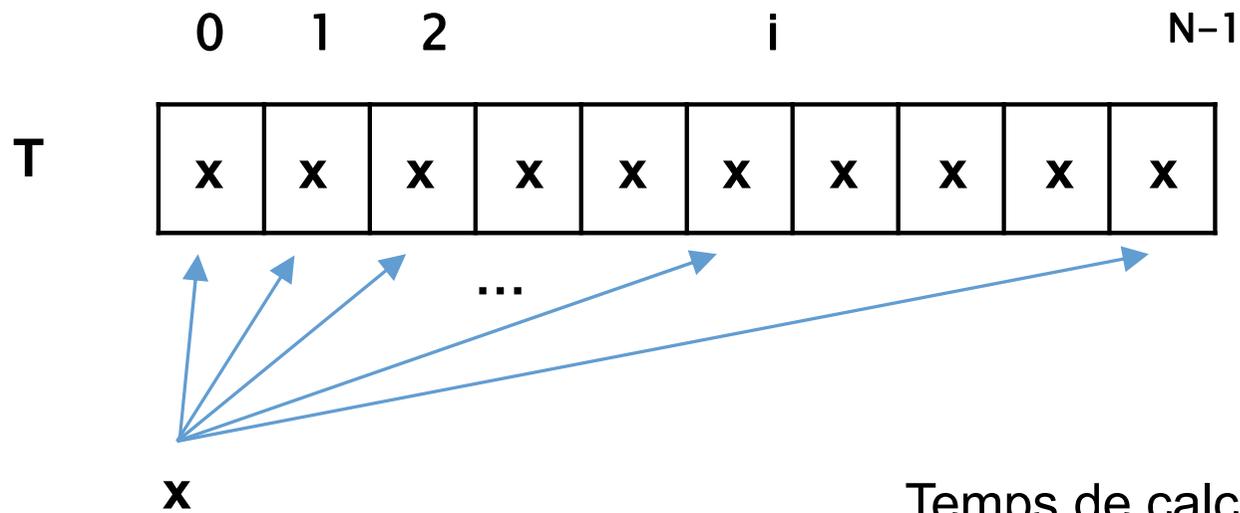
Dans quel ordre se  
remplit le tableau ?

	0	1	2
0			
1			
2			
3			
4			

# Tableaux : activités

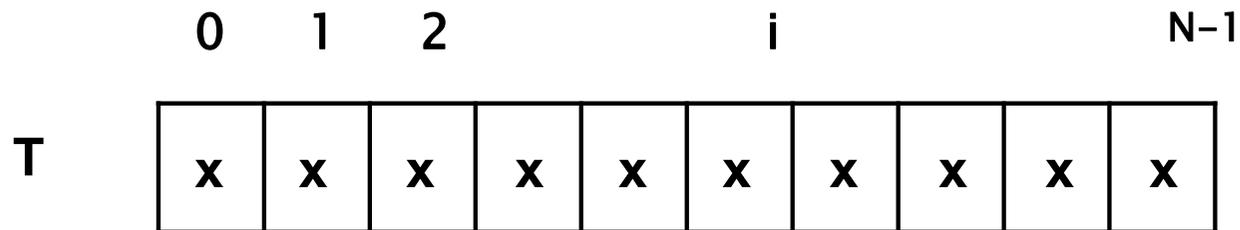
- ▶ Initialiser un tableau.
- ▶ Afficher les éléments d'un tableau.
- ▶ Rechercher si une valeur est dans un tableau.
- ▶ Échanger le contenu de 2 cases du tableau.
- ▶ Insérer un nouvel élément dans un tableau.
- ▶ Pour tous les éléments du tableau faire...
- ▶ Existe-t-il un élément du tableau qui...?
- ▶ **Trier un tableau.**

# Initialiser un tableau



Temps de calcul ?

# Initialiser un tableau



En entrée :

**T** tableau de **N** entiers

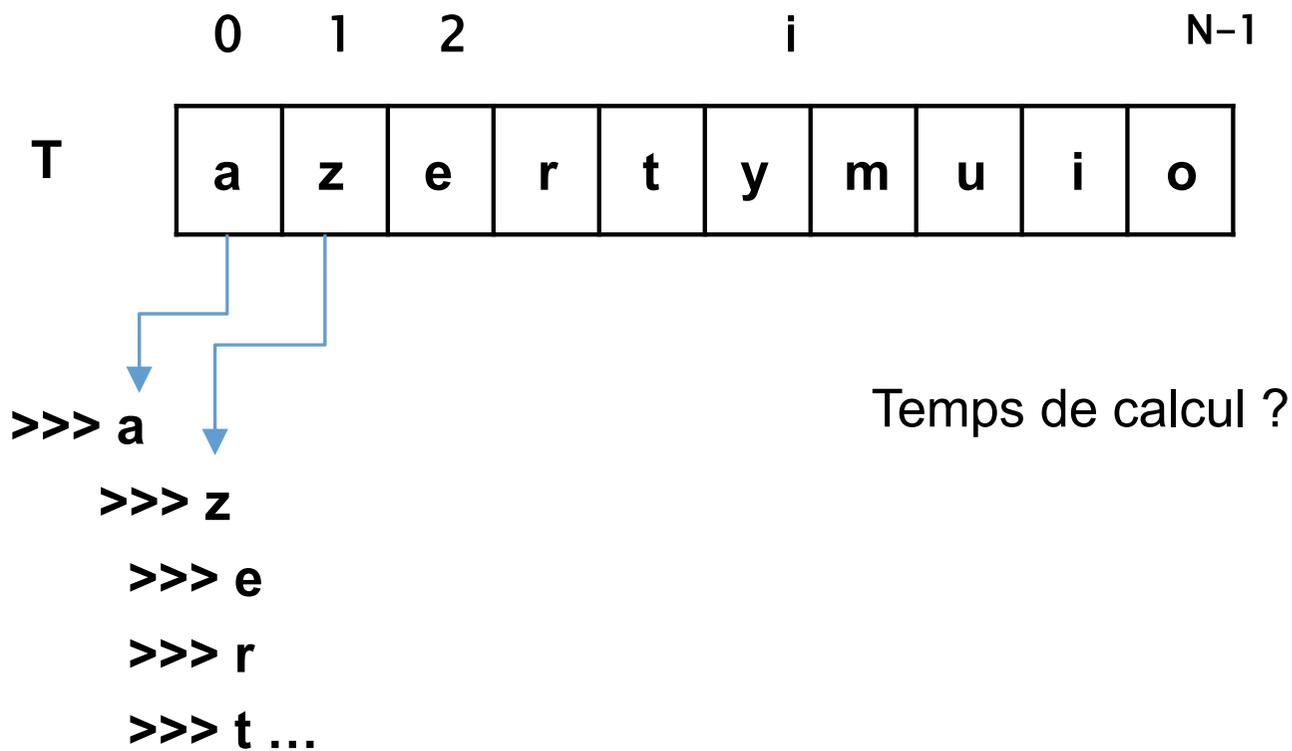
**x** un entier

```
for i in range(N) :
```

```
    T[i]=x
```

En sortie :

# Afficher les éléments d'un tableau



# Afficher les éléments d'un tableau

	0	1	2		i				N-1	
T	a	z	e	r	t	y	m	u	i	o

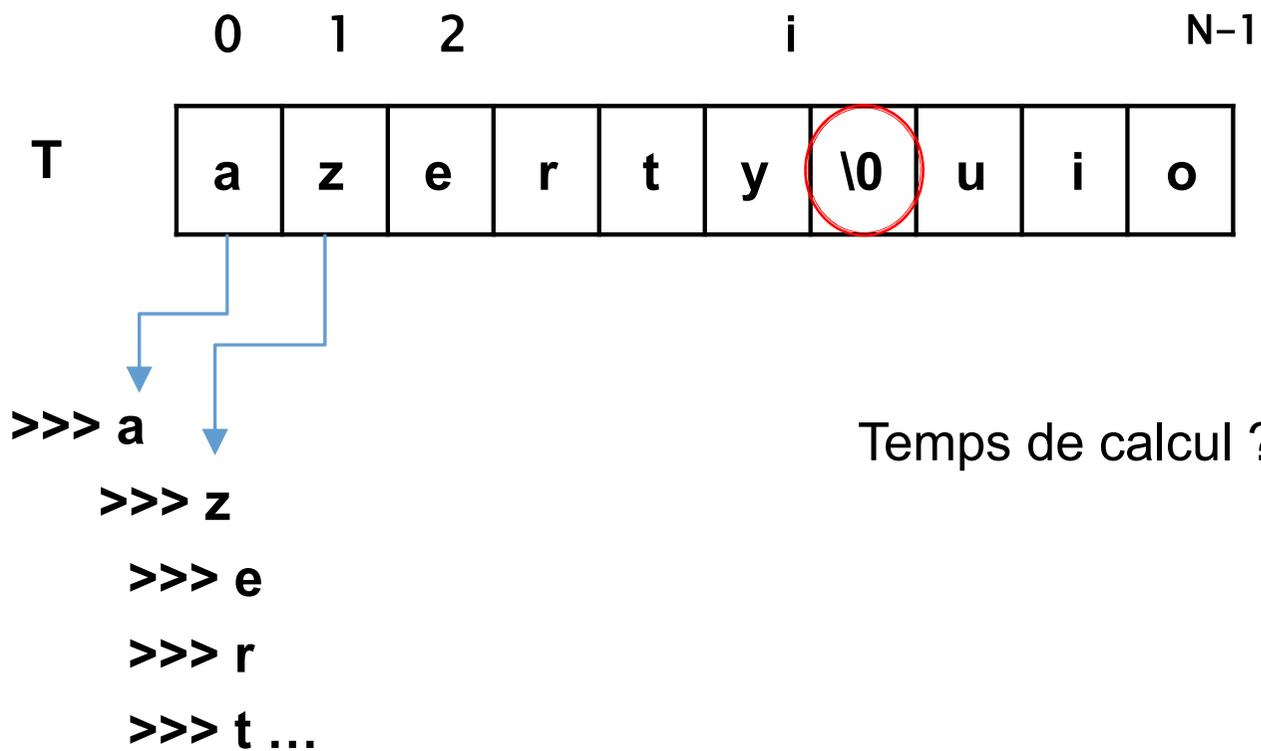
En entrée :

**T** tableau de **N** caractères

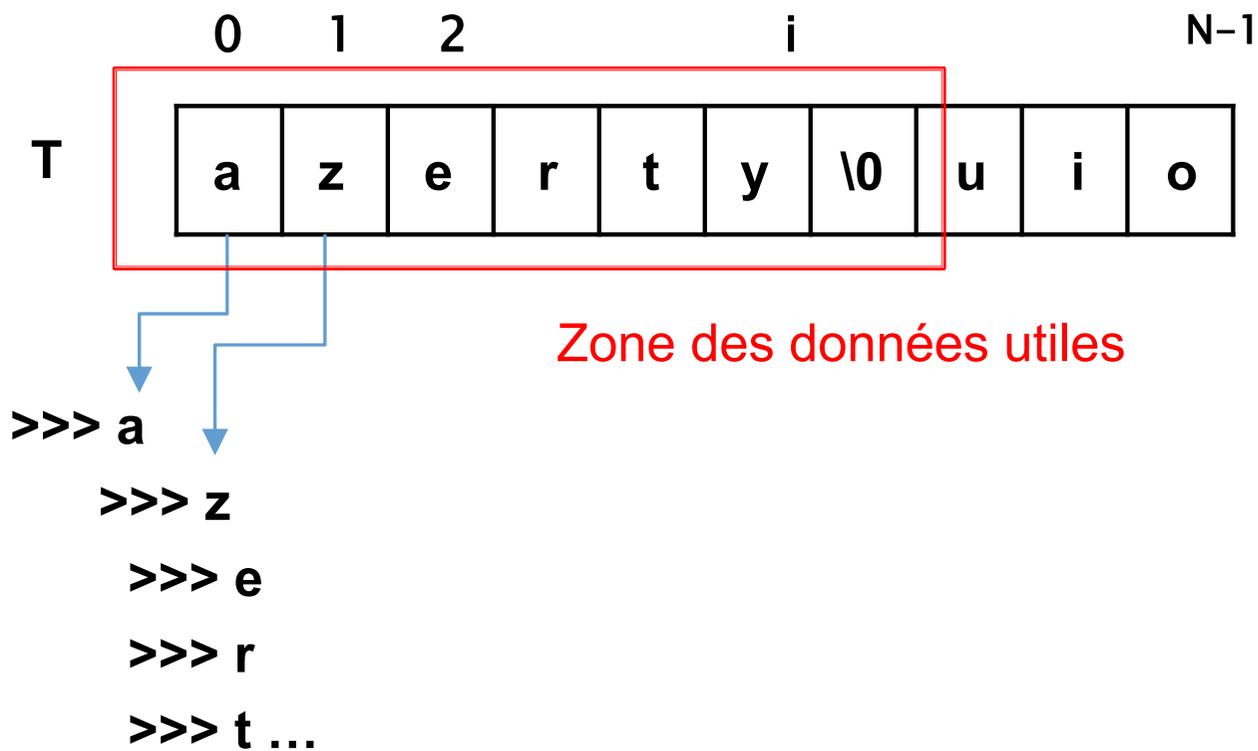
```
for i in range(N):  
    print(T[i])
```

En sortie :

# Afficher les éléments d'un tableau



# Afficher les éléments d'un tableau



# Afficher les éléments d'un tableau

	0	1	2		i				N-1	
T	a	z	e	r	t	y	\0	u	i	o

En entrée :

**T** tableau de caractères

**\0** indique la fin de chaîne

**i=0**

```
while(i<N and T[i]!='\0'):
```

```
    print(T[i])
```

```
    i=i+1
```

En sortie :

# Rechercher un élément dans un tableau

	0	1	2		i				N-1	
T	a	z	e	r	t	y	\0	u	i	o
	?	?	?	?	?					

Temps de calcul ?

t ?

# Rechercher un élément dans un tableau

	0	1	2		i				N-1	
T	a	z	e	r	t	y	\0	u	i	o

En entrée :

**T** tableau de caractères

**x** le caractère recherché

**i=0**

```
while (i < N and T[i] != '\0') :
```

```
    if T[i] == x :
```

```
        return True
```

```
    i=i+1
```

```
return False
```

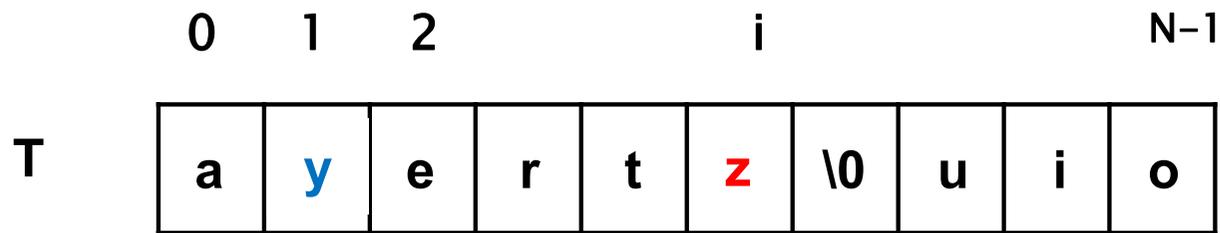
En sortie :

Booléen retourné

**T** n'est pas modifié

Amélioration ?

# Echange le contenu de 2 cases dans un tableau



tmp



Temps de calcul ?



# Echange le contenu de 2 cases dans un tableau

	0	1	2		i				N-1	
T	a	y	e	r	t	z	\0	u	i	o

En entrée :

T tableau de caractères

2 entiers i et j

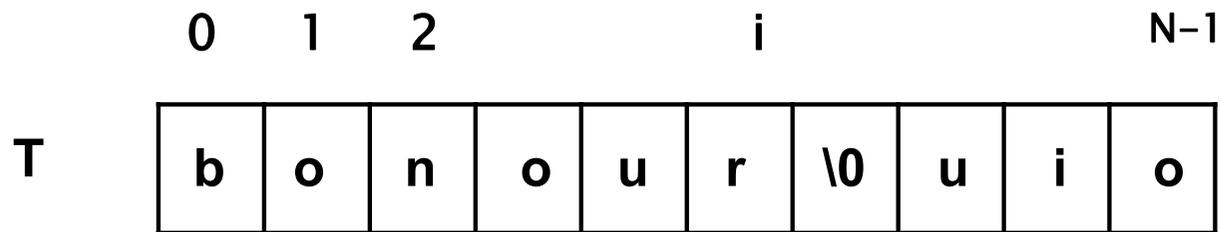
Tmp=T[i]

T[i]=T[j]

T[j]=Tmp

En sortie :

# Insérer un élément dans un tableau

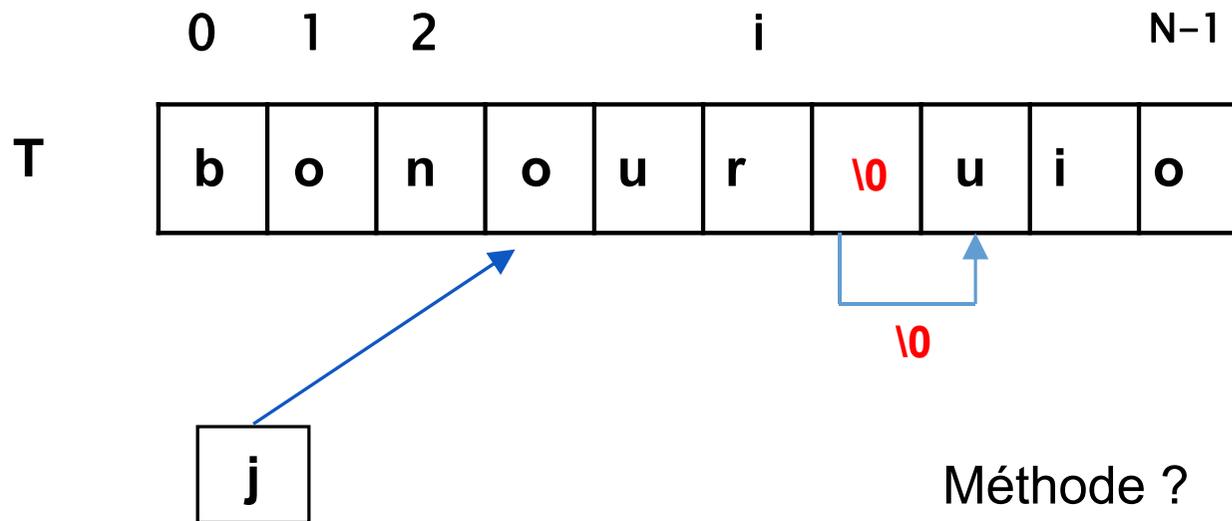


j

Méthode ?

Temps de calcul ?

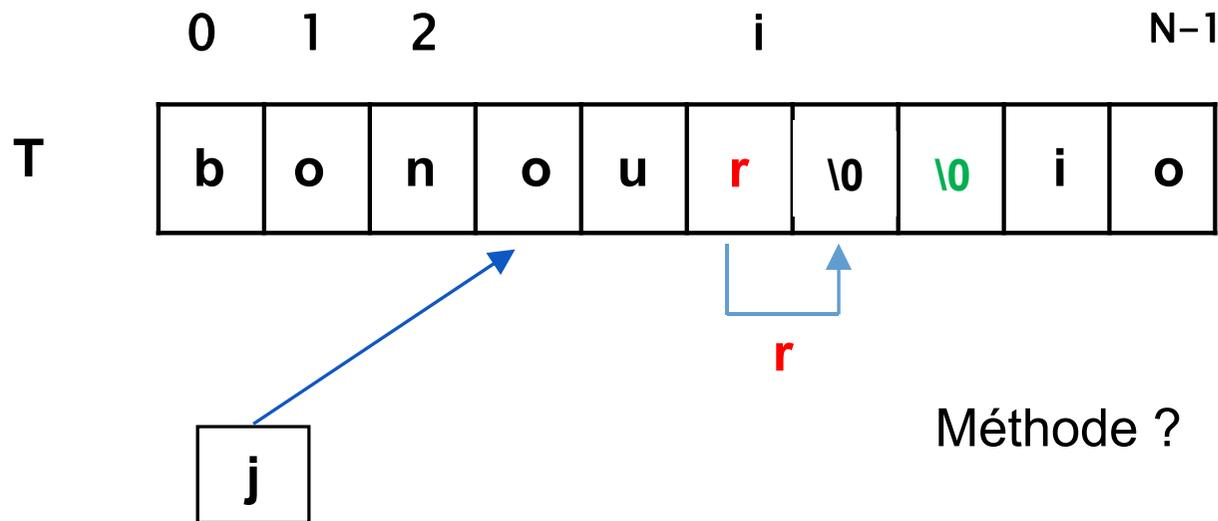
# Insérer un élément dans un tableau



Méthode ?

Temps de calcul ?

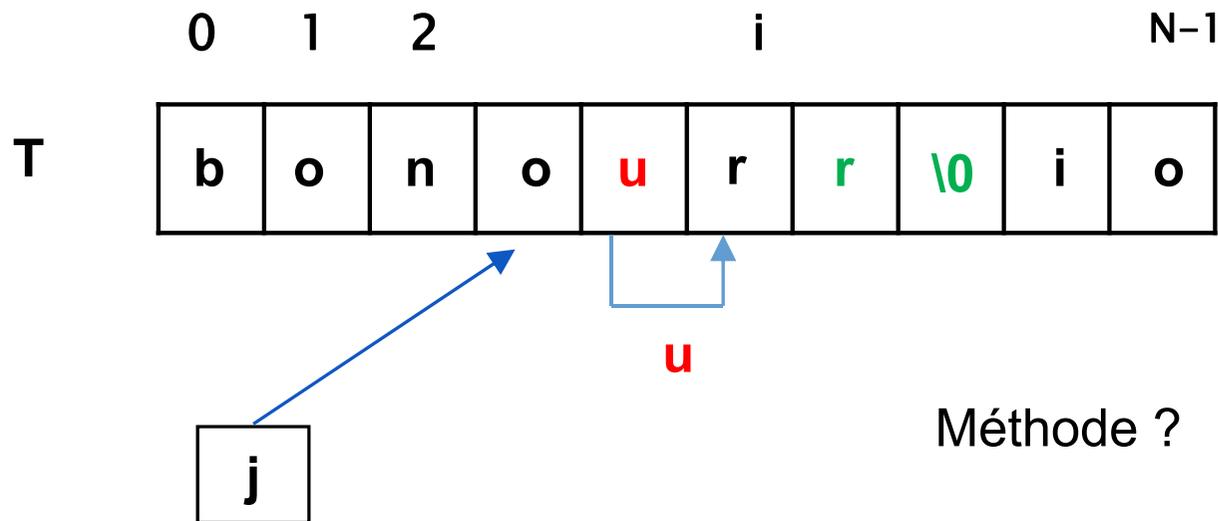
# Insérer un élément dans un tableau



Méthode ?

Temps de calcul ?

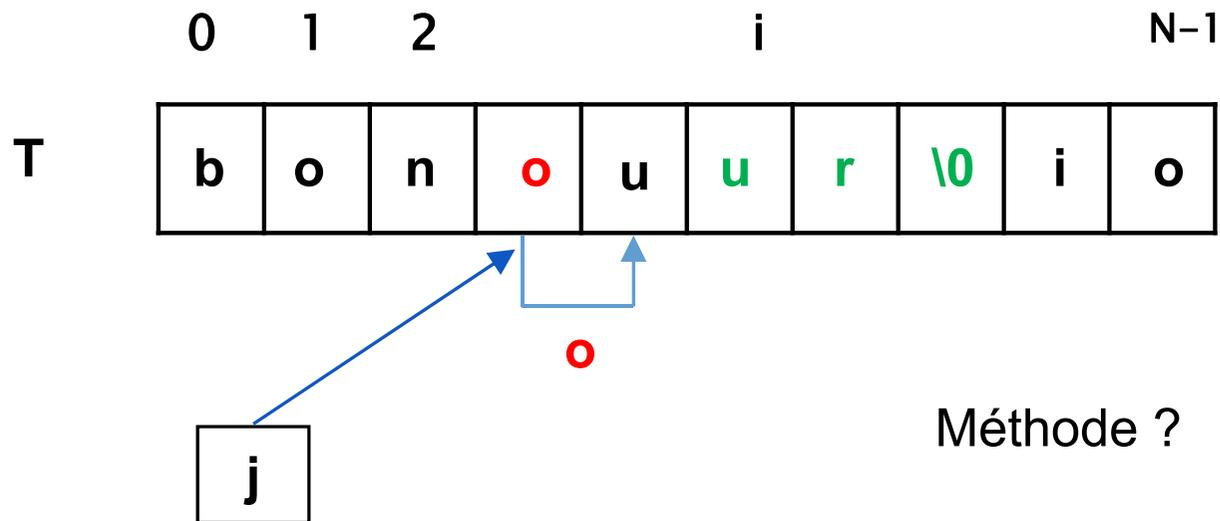
# Insérer un élément dans un tableau



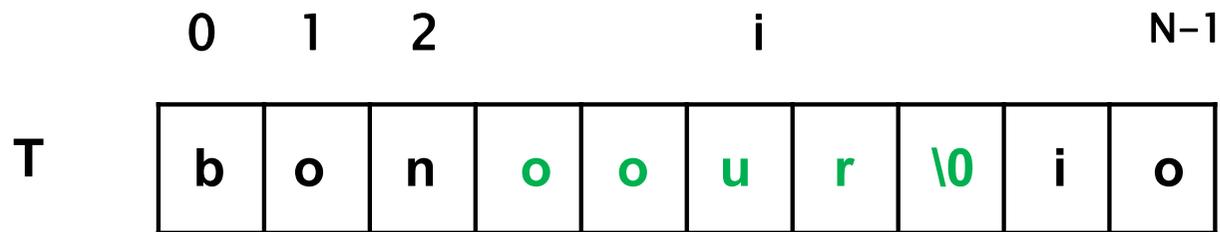
Méthode ?

Temps de calcul ?

# Insérer un élément dans un tableau



# Insérer un élément dans un tableau

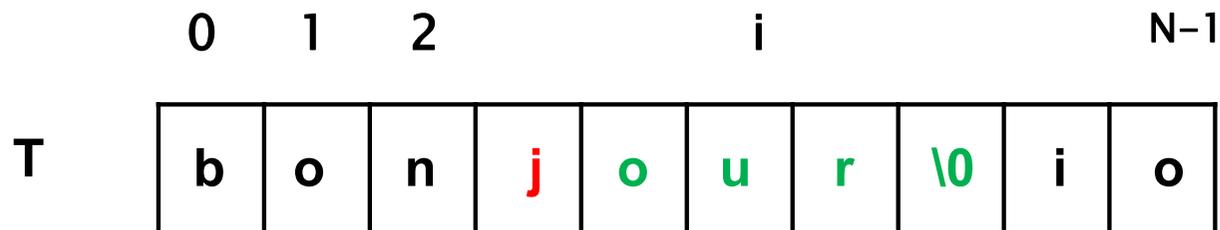


j

Méthode ?

Temps de calcul ?

# Insérer un élément dans un tableau

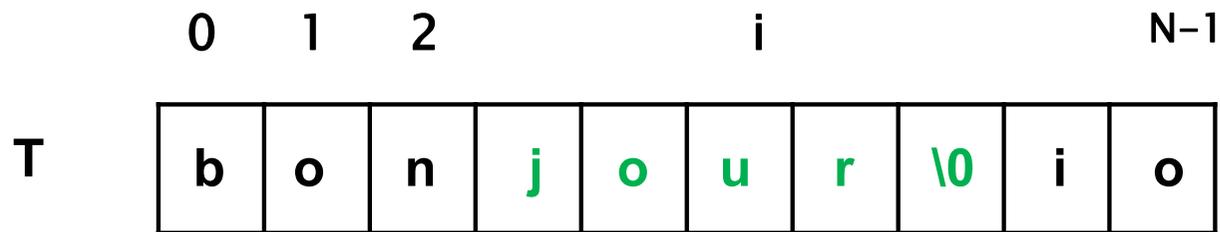


j

Méthode ?

Temps de calcul ?

# Insérer un élément dans un tableau



j

Méthode ?

Temps de calcul ?

# Trier un tableau

	0	1	2		i				N-1	
T	b	o	n	o	u	r	\0	u	i	o

Méthode ?

Temps de calcul ?

# Exercice

```
def quesako (T, n) :  
    i=0  
    while (i<n and T[i]>0) :  
        i=i+1  
    return (i==n)
```

# Exercice

```
def quesako (T, n) :  
    i=0  
    while (i<n and T[i]>0) :  
        i=i+1  
    if (i==n) :  
        return True  
    return False
```