

Tri par insertion

Principe

Soit T un tableau d'entiers. Le tri par insertion consiste à :

- 1). L'élément à la position 0 est dans sa bonne place.
- 2). On commence à partir du second élément (qui se trouve à la position = 1)
- 3). à l'étape i , on cherche la bonne position de l'élément $T[i]$ dans le sous tableau trié de 0 à $i-1$.

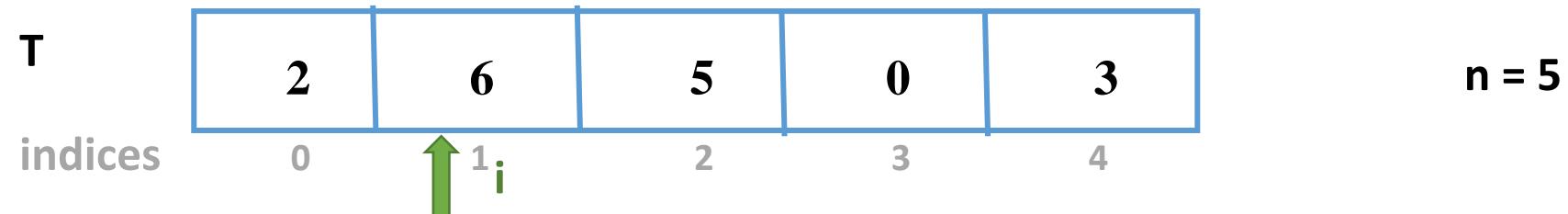
Code C de l'algorithme tri par insertion

```
for (i=1;i<n;i++)
{
    tmp = T[i]; // mémoriser T[i] dans une variable tmp
    j = i;

    // décaler vers la droite les éléments de T[0]..T[i-1] qui sont plus grands que tmp en
    // partant de T[i-1]
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



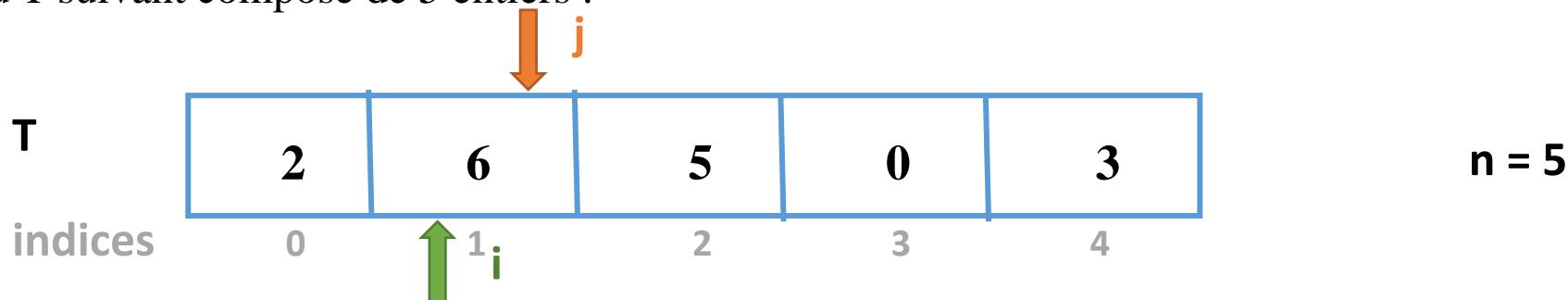
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
6

i = 1

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :

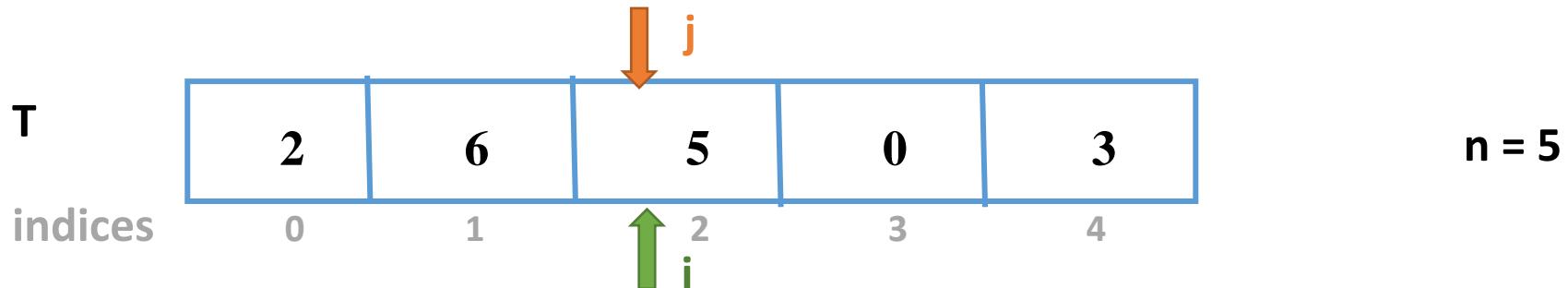


```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp )) → while (( 1>0 ) && (2 > 6 ) ) --> FAUX
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
6

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



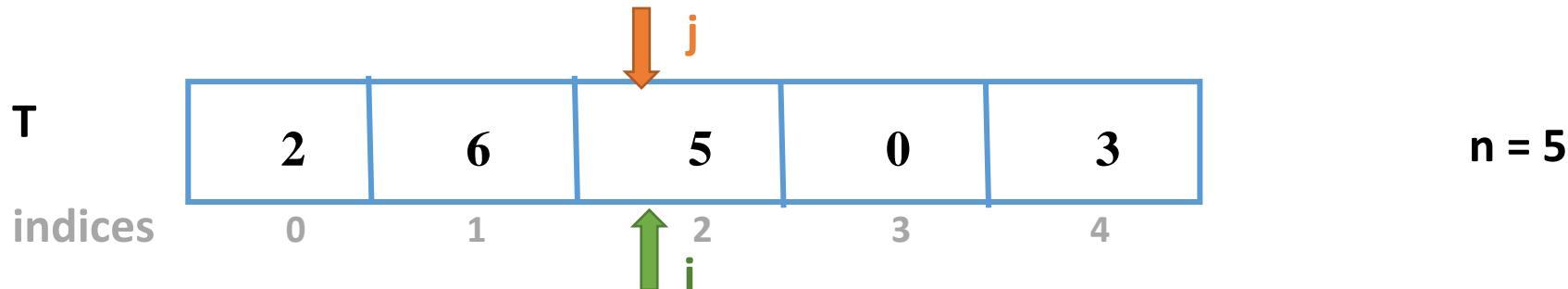
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
5

while ((2 > 0) && (6 > 5)) --> VRAI
i = 2

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
5

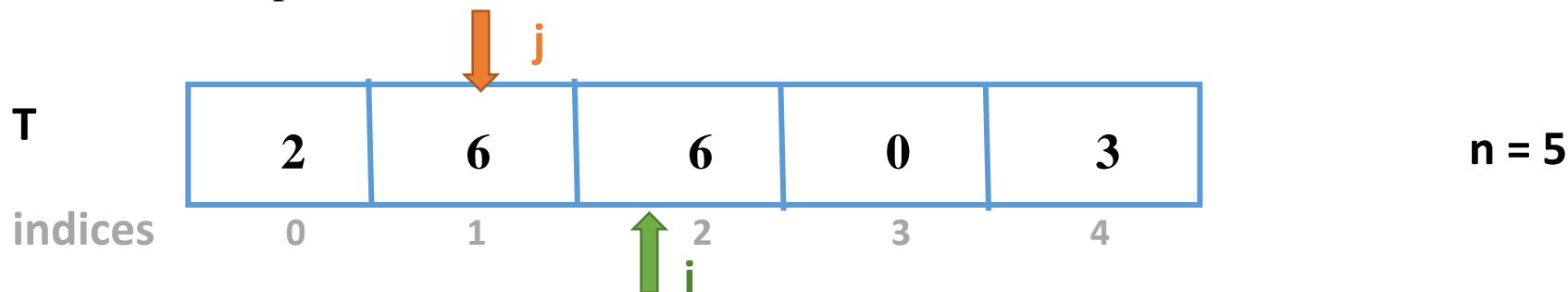
while ((2 > 0) && (6 > 5)) --> VRAI

{
 T[2] = T[1];
 j = 1 ;
}
T[j] = 5 ;

i = 2

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)  
{  
    tmp = T[i];  
    j = i;  
    while(j > 0 && (T[j-1] > tmp ))  
    {  
        T[j] = T[j-1];  
        j--;  
    }  
    T[j] = tmp;  
}
```

tmp
5

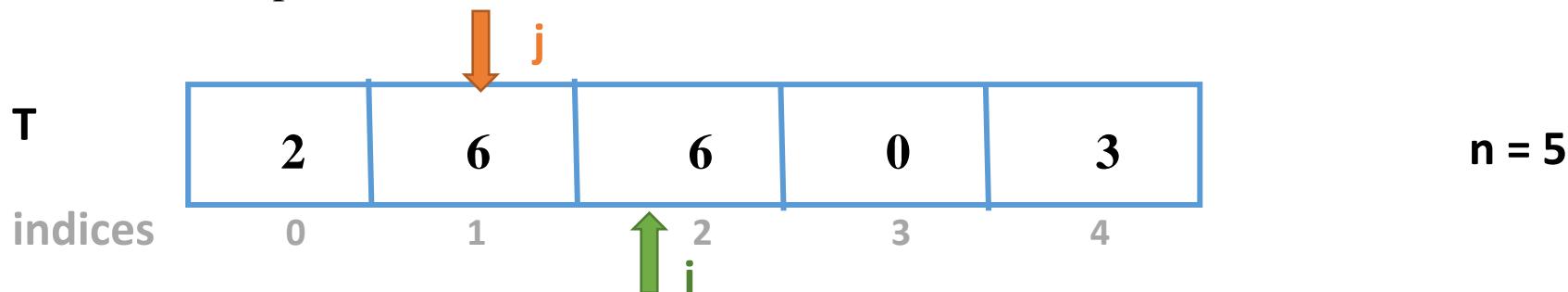
while ((2 > 0) && (6 > 5)) --> VRAI

{
 T[2] = T[1];
 j = 1 ;
}
T[j] = 5 ;

i = 2

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

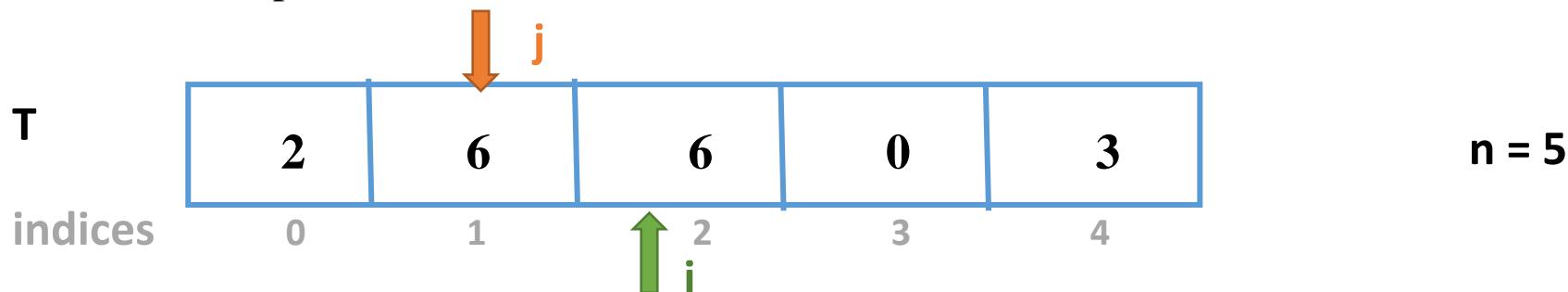
$\text{tmp} = 5$
 $i = 2$

$\text{while } ((1 > 0) \&\& (2 > 5)) \rightarrow \text{FAUX}$

$\{$
 $\quad T[2] = T[1];$
 $\quad j = 1;$
 $\}$
 $T[j] = 5 ;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

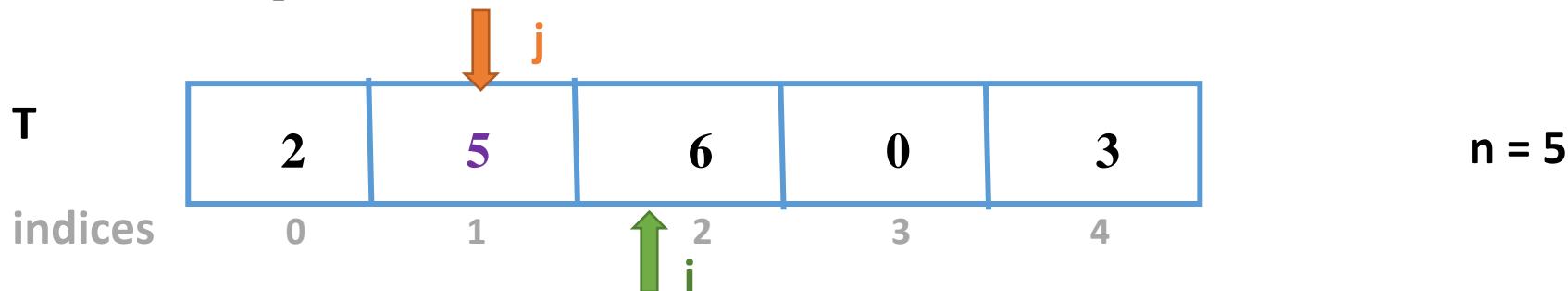
tmp
5

while ((1 > 0) && (2 > 5)) --> FAUX

{
 T[2] = T[1];
 j = 1 ;
}
T[1] = 5 ;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

$\xrightarrow{\hspace{1cm}}$

$\text{tmp} = 5$

$i = 2$

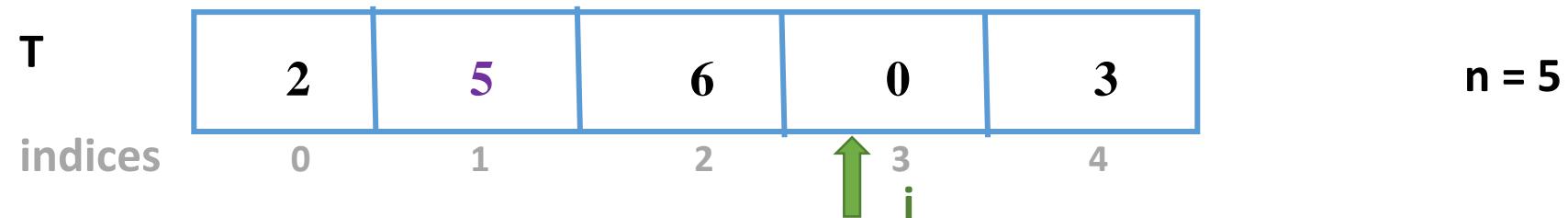
$\text{while } ((1 > 0) \&\& (2 > 5)) \rightarrow \text{FAUX}$

{
 $T[2] = T[1];$
 $j = 1;$
}

$T[1] = 5;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



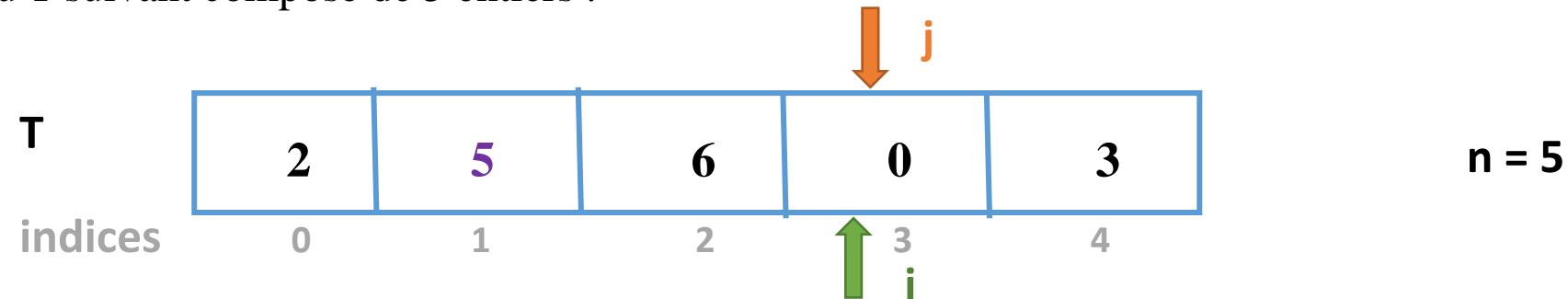
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



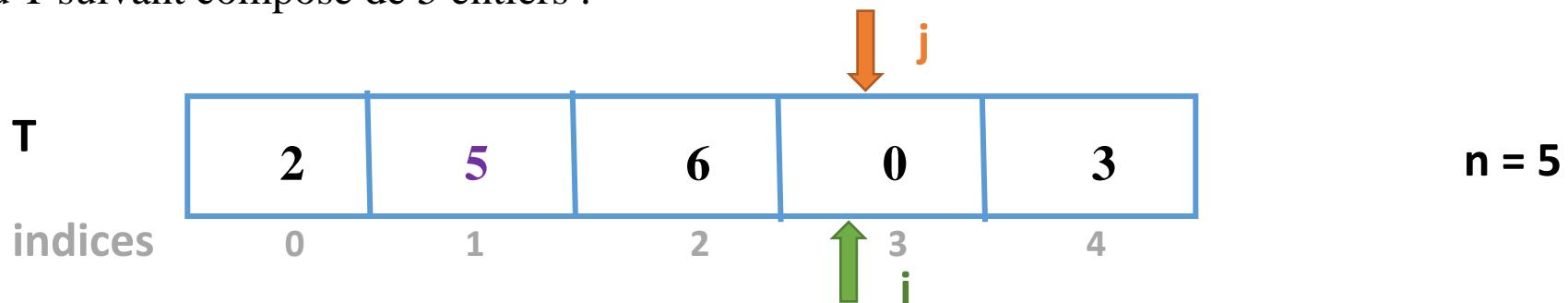
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)  
{  
    tmp = T[i];  
    j = i;  
    while(j > 0 && (T[j-1] > tmp ))  
    {  
        T[j] = T[j-1];  
        j--;  
    }  
    T[j] = tmp;  
}
```

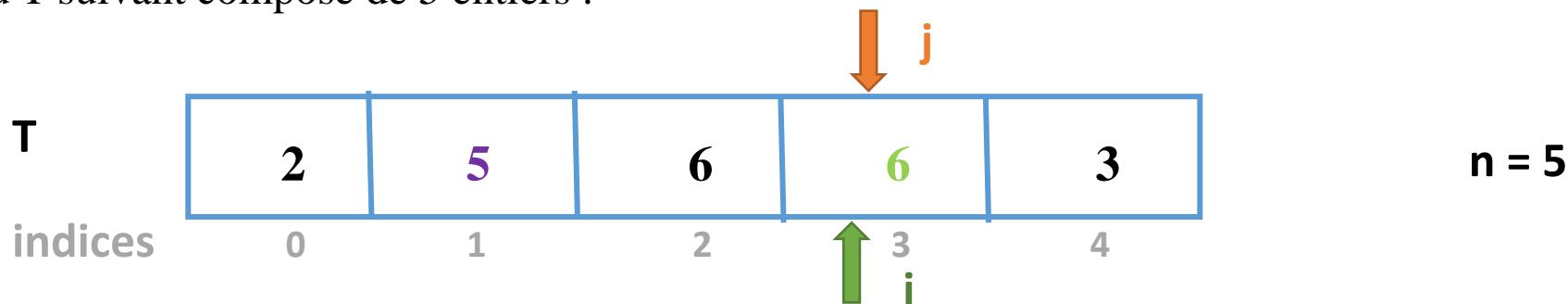
tmp
0

while ((3 > 0) && (6 > 0)) --> VRAI

{
 T[3] = T[2];
 j = 2 ;
}
T[j] = 0 ;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

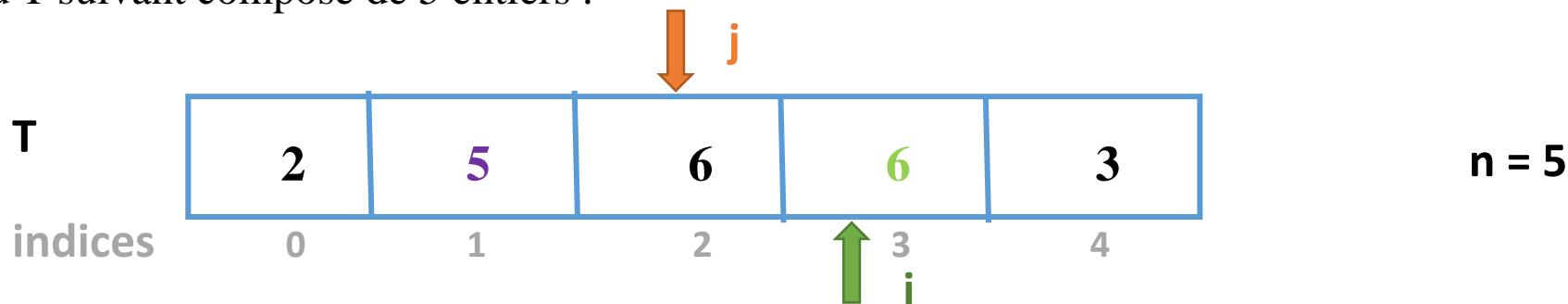
while ((3 > 0) && (6 > 0)) --> VRAI

{
 T[3] = T[2];
 j = 2 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)  
{  
    tmp = T[i];  
    j = i;  
    while(j > 0 && (T[j-1] > tmp ))  
    {  
        T[j] = T[j-1];  
        j--;  
    }  
    T[j] = tmp;  
}
```

tmp
0

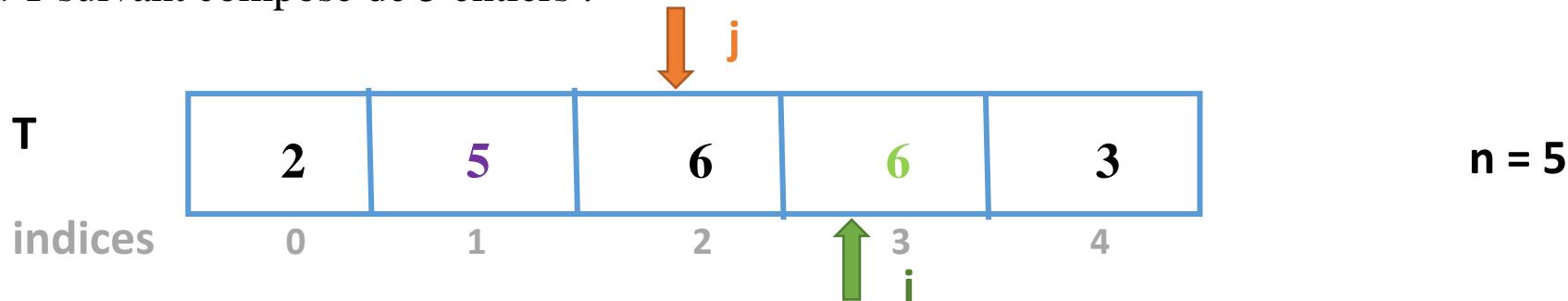
while ((3 > 0) && (6 > 0)) --> VRAI

{
 T[3] = T[2];
 j = 2 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

i = 3

while ((3 > 0) && (6 > 0)) --> **VRAI**

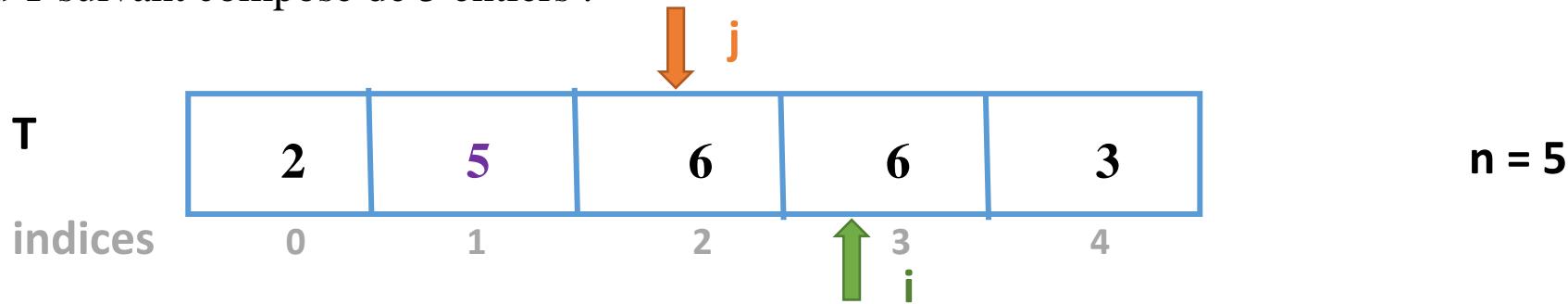
{
 T[3] = T[2];
 j = 2;
}

T[j] = 0 ;

A blue arrow points from the code line "while(j > 0 && (T[j-1] > tmp))" to the start of the loop body. A large grey curved arrow indicates the loop iteration, starting from the condition and pointing back to the top of the loop body.

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

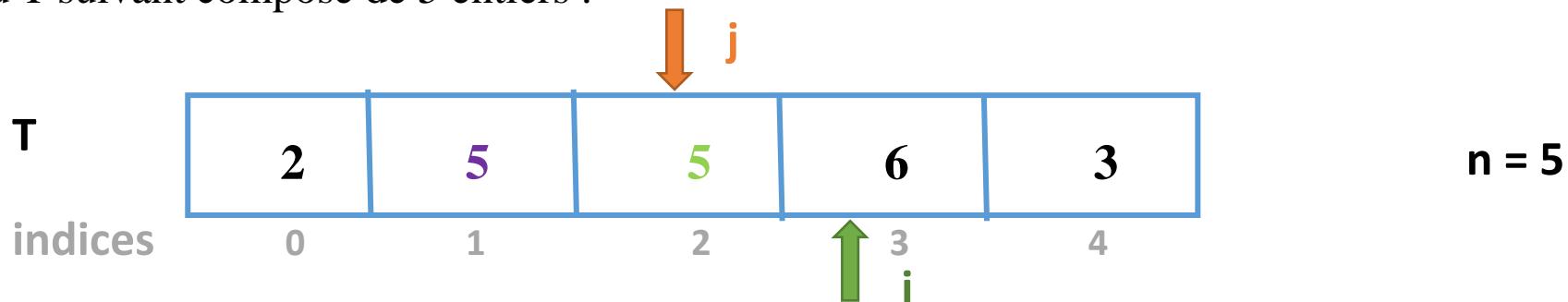
while ((2 > 0) && (5 > 0)) --> VRAI

{
 T[2] = T[1];
 j = 1 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

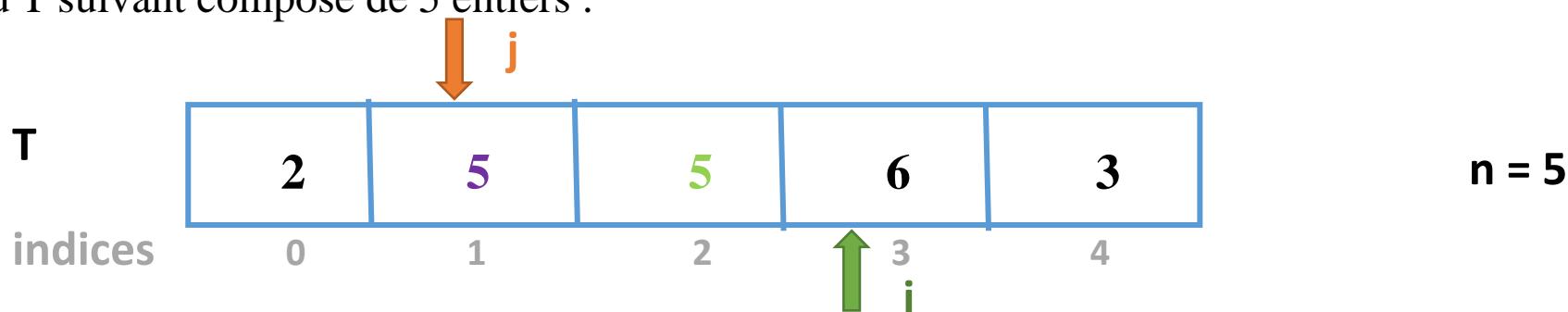
while ((2 > 0) && (5 > 0)) --> VRAI

{
 T[2] = T[1];
 j = 1 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

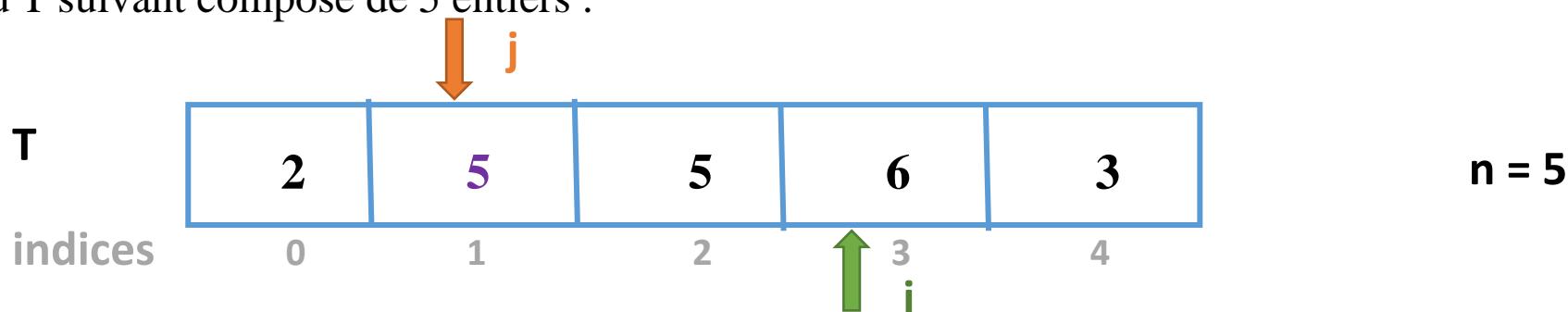
while ((2 > 0) && (5 > 0)) --> VRAI

{
 T[2] = T[1];
 j = 1 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

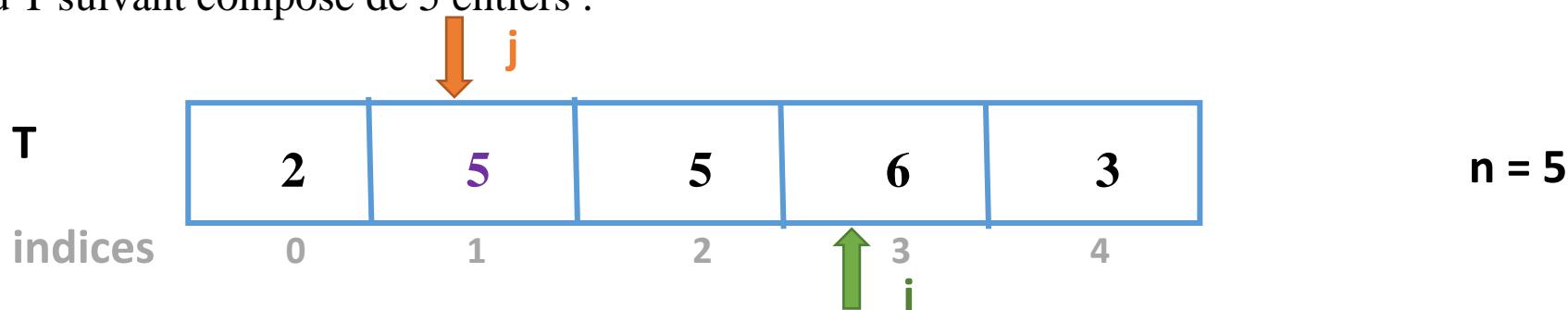
while ((2 > 0) && (5 > 0)) --> VRAI

{
 T[2] = T[1];
 j = 1;
}
T[j] = 0 ;

A blue arrow points from the condition in the while loop to the text "VRAI". A large curved grey arrow indicates the flow of execution from the end of the while loop back to the start of the loop body.

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

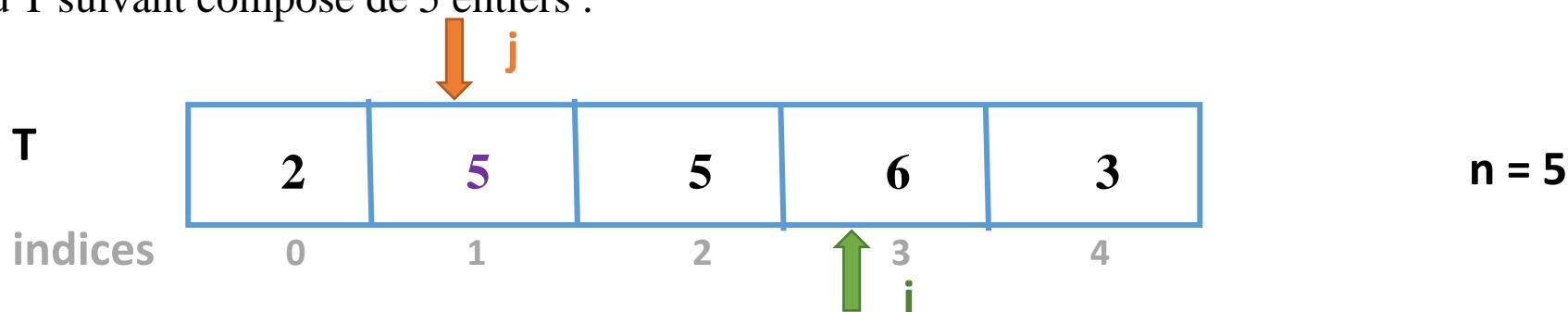
tmp
0

while ((1 > 0) && (2 > 0))
{
 T[1] = T[0];
 j = 0 ;
}
T[j] = 0 ;

$i = 3$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

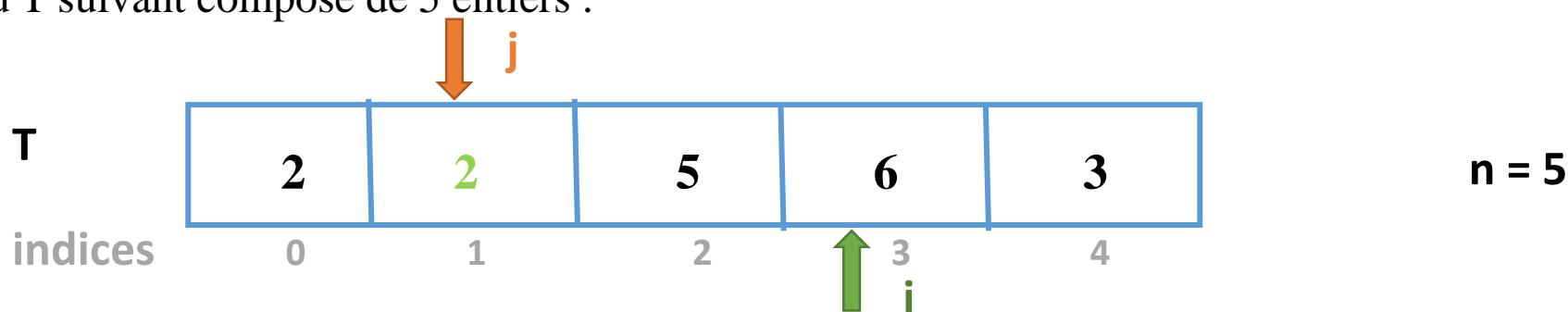
tmp
0

while ((1 > 0) && (2 > 0)) --> VRAI

{
 T[1] = T[0];
 j = 0 ;
}
T[j] = 0 ;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)  
{  
    tmp = T[i];  
    j = i;  
    while(j > 0 && (T[j-1] > tmp ))  
    {  
        T[j] = T[j-1];  
        j--;  
    }  
    T[j] = tmp;  
}
```

tmp
0

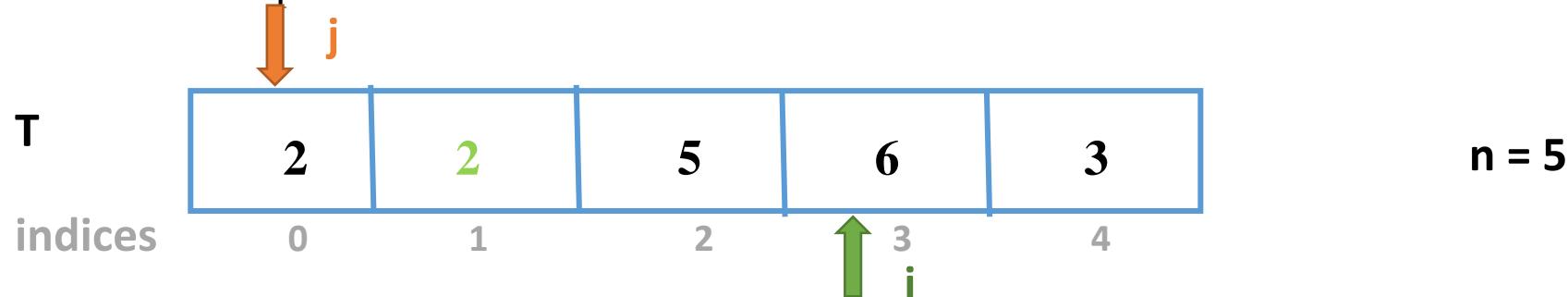
while ((1 > 0) && (2 > 0)) --> VRAI

{
 T[1] = T[0];
 j = 0 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

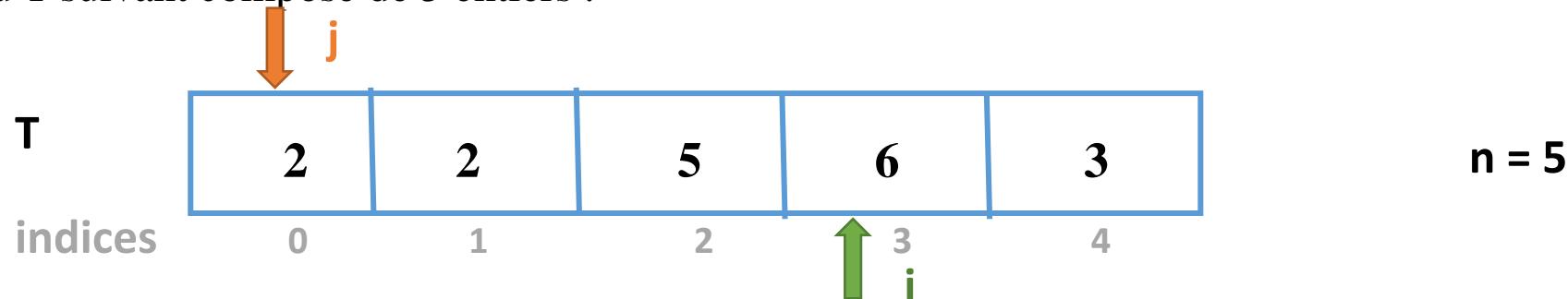
while ((1 > 0) && (2 > 0)) --> VRAI

{
 T[1] = T[0];
 j = 0 ;
}
T[j] = 0 ;

i = 3

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

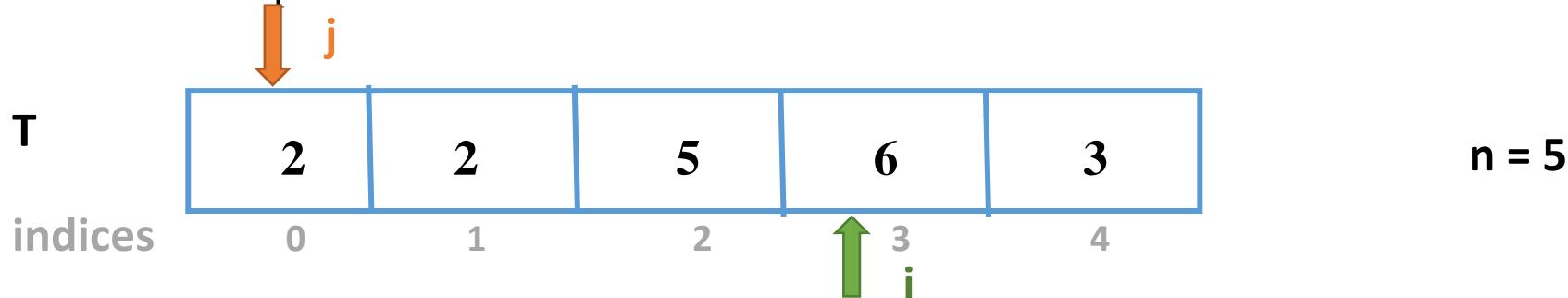
tmp
0

while (**(0 > 0) &&**) --> FAUX

{
}
 $T[0] = 0;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

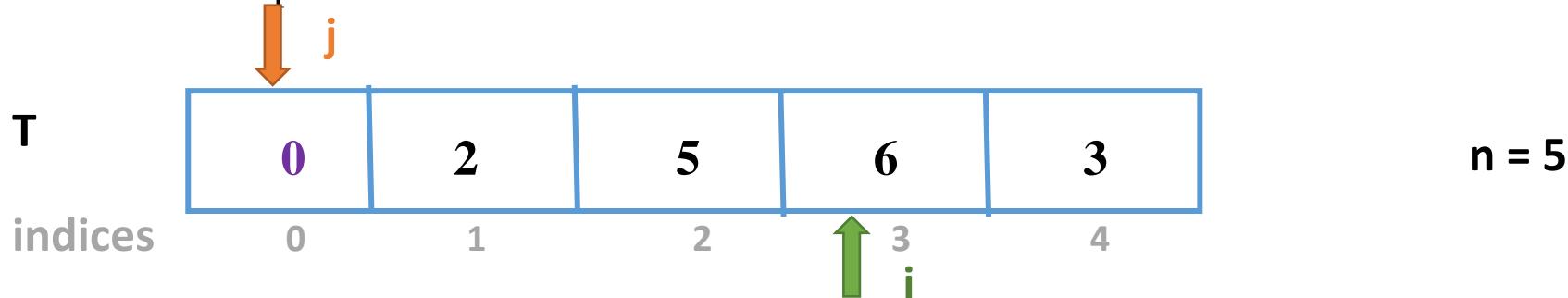
tmp
0

while (**(0 > 0) &&**) --> FAUX

{
}
T[0] = 0;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
0

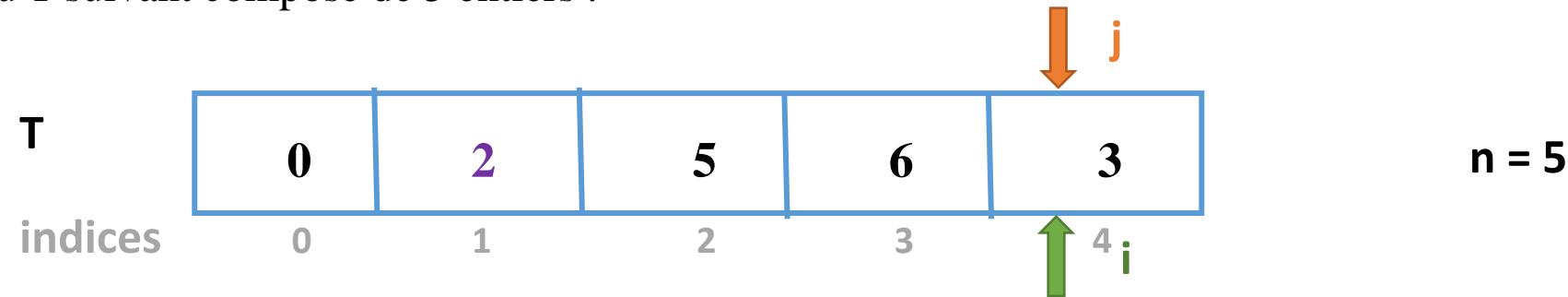
while (**(0 > 0) &&**) --> FAUX

{
}
T[0] = 0;

The code snippet shows a bubble sort algorithm. It initializes i to 1, tmp to $T[i]$, and j to i . It then enters a while loop where it compares $T[j-1]$ and tmp . Since $0 > 0$ is false, the loop exits. The value 0 is then assigned to $T[0]$.

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



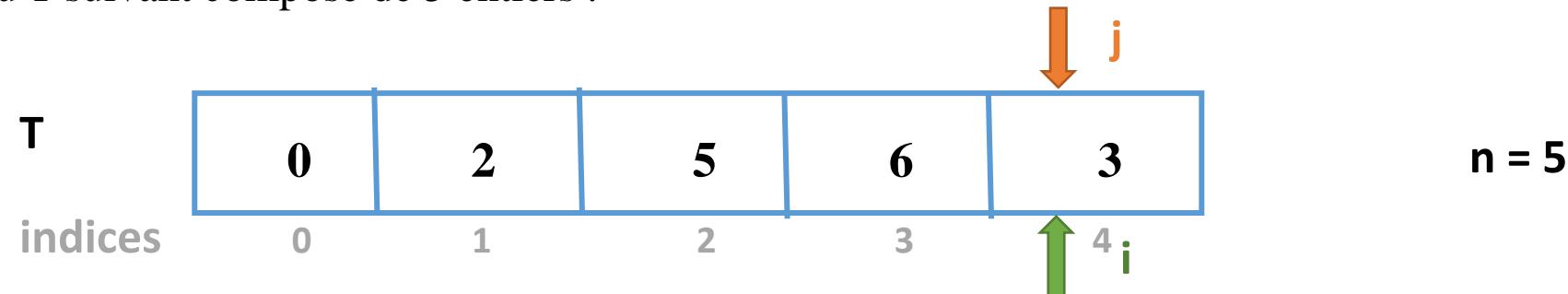
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3

i = 4

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :

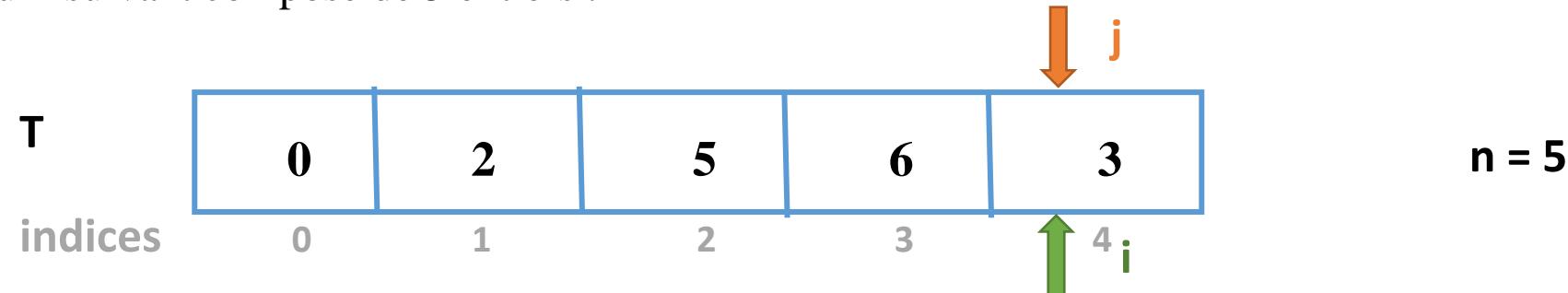


```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3
while ((4 > 0) && (6 > 3))
i = 4

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))    —————→ while (( 4 > 0 ) && (6 > 3 ) ) --> VRAI
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

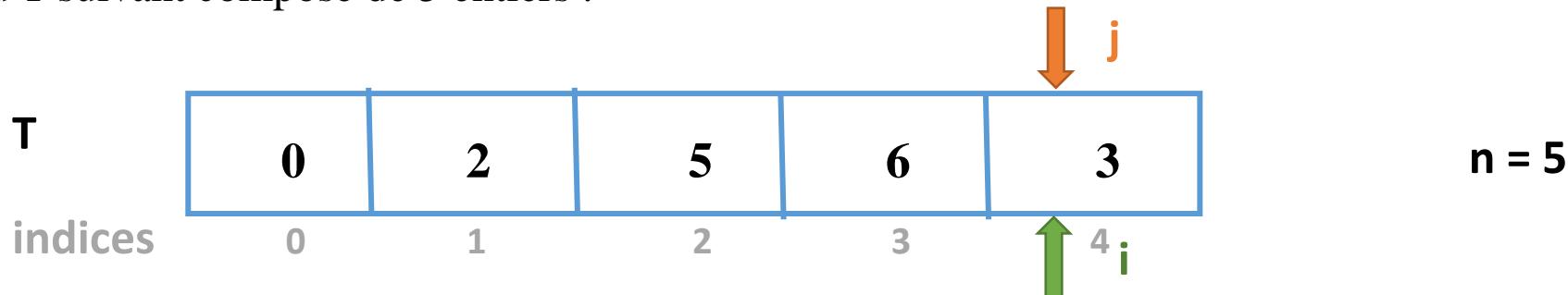
tmp

3

i = 4

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

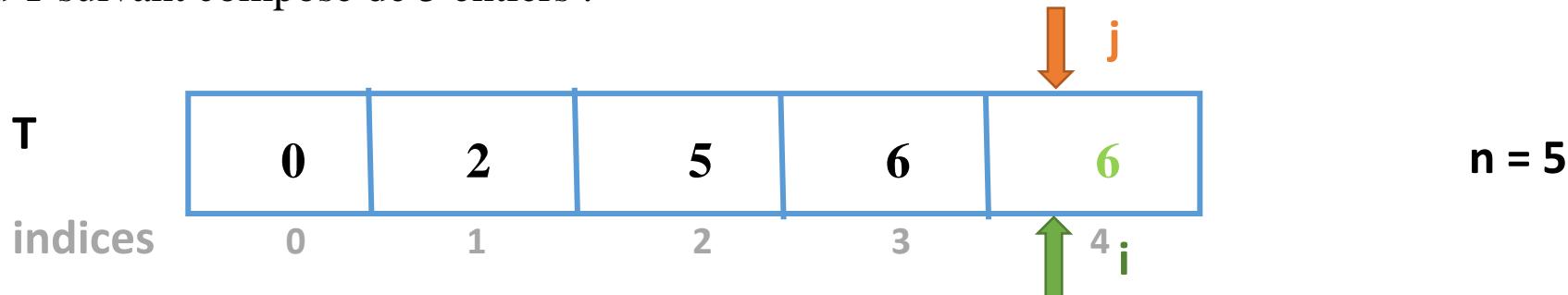
tmp
3
while ((4 > 0) && (6 > 3)) --> VRAI

```
{
    T[4] = T[3];
    j = 3;
}
```

$T[j] = 3;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3

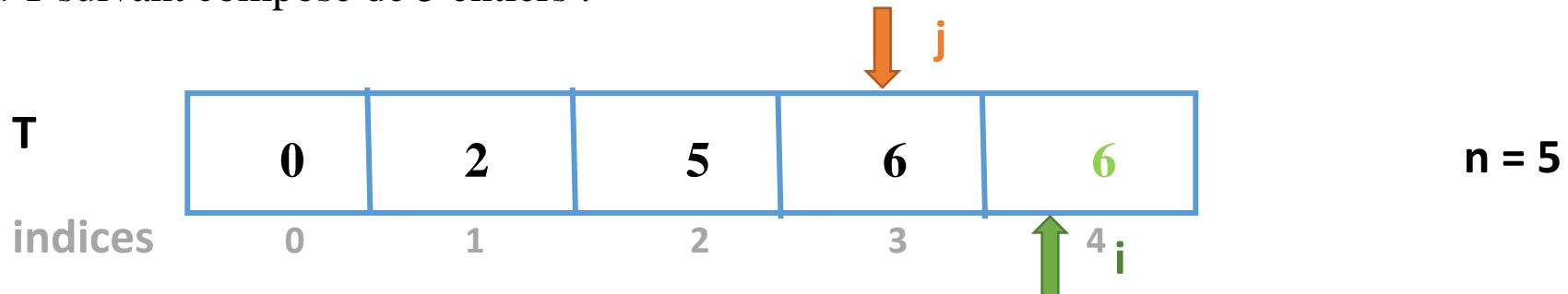
while ((4 > 0) && (6 > 3)) --> VRAI

```
{
    T[4] = T[3];
    j = 3;
}
```

T[j] = 3;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

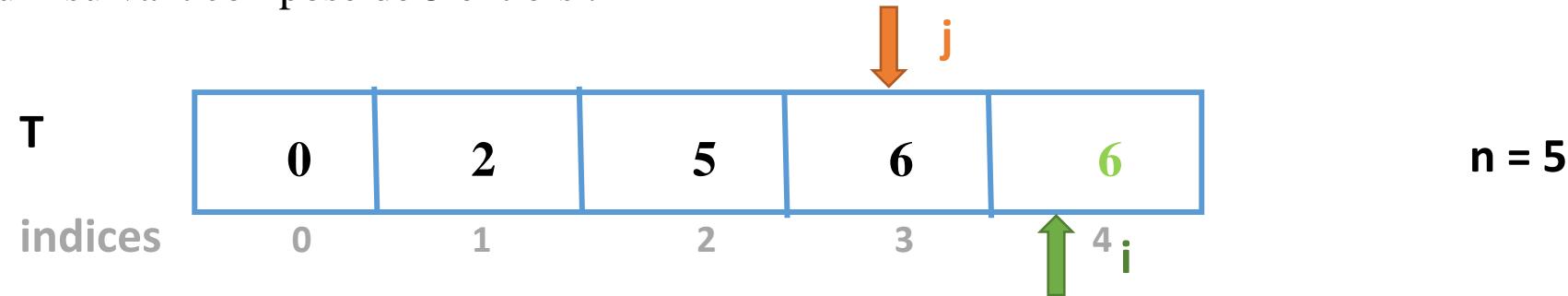
tmp
3
while ((4 > 0) && (6 > 3)) --> VRAI

```
{
    T[4] = T[3];
    j = 3;
}
```

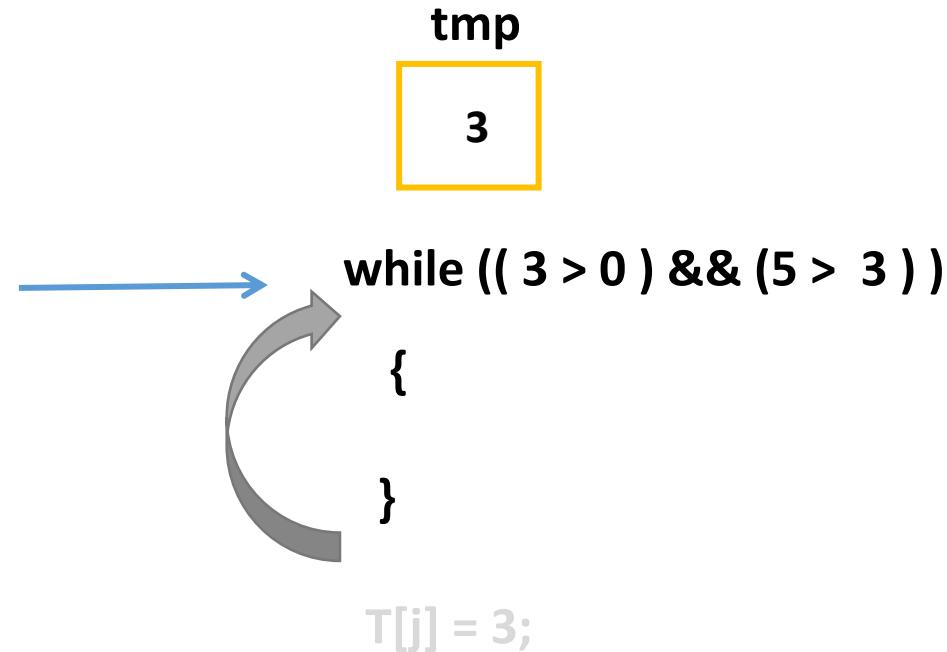
$T[j] = 3;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :

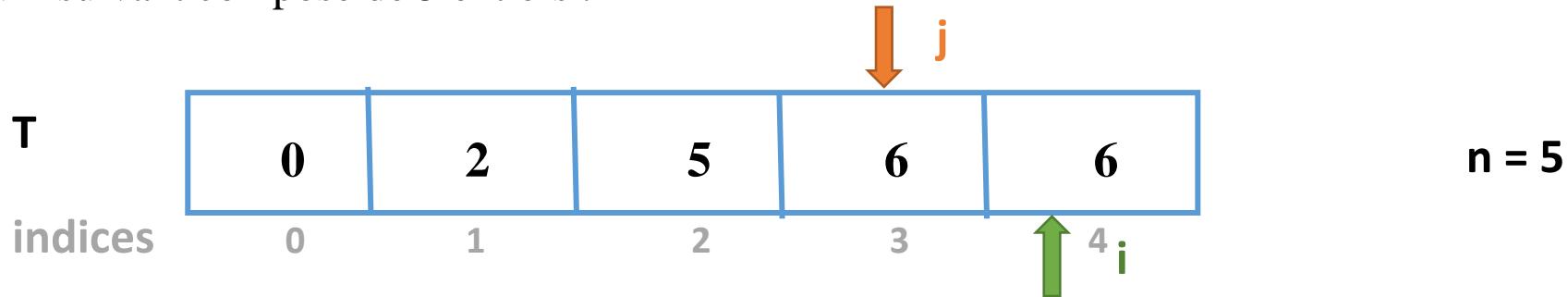


```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```



Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3

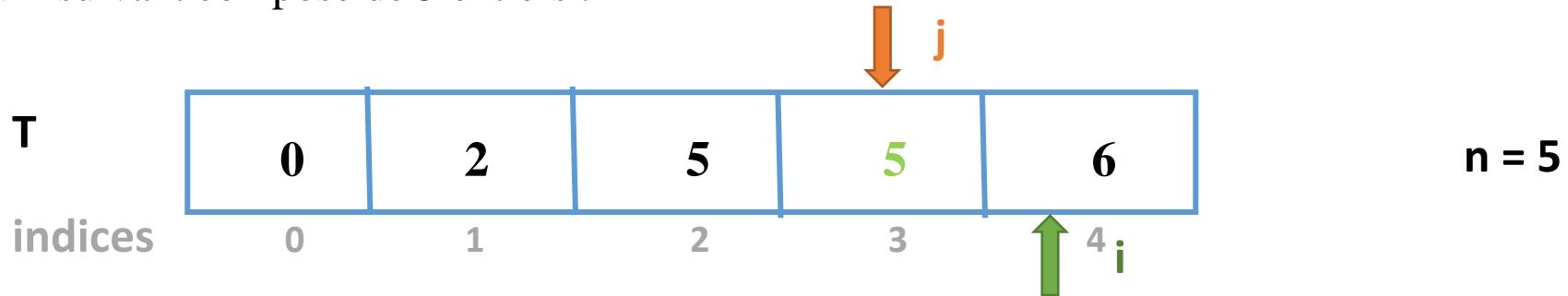
$i = 4$

while ((3 > 0) && (5 > 3)) --> VRAI

{
}
T[j] = 3;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

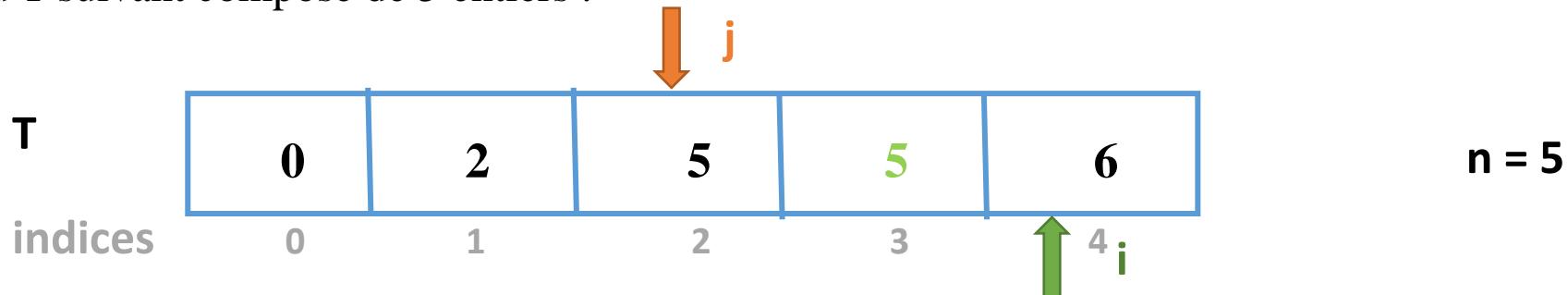
tmp
3
while ((3 > 0) && (5 > 3)) --> VRAI

```
{
    T[3] = T[2];
    j= 2;
}
```

$T[j] = 3;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

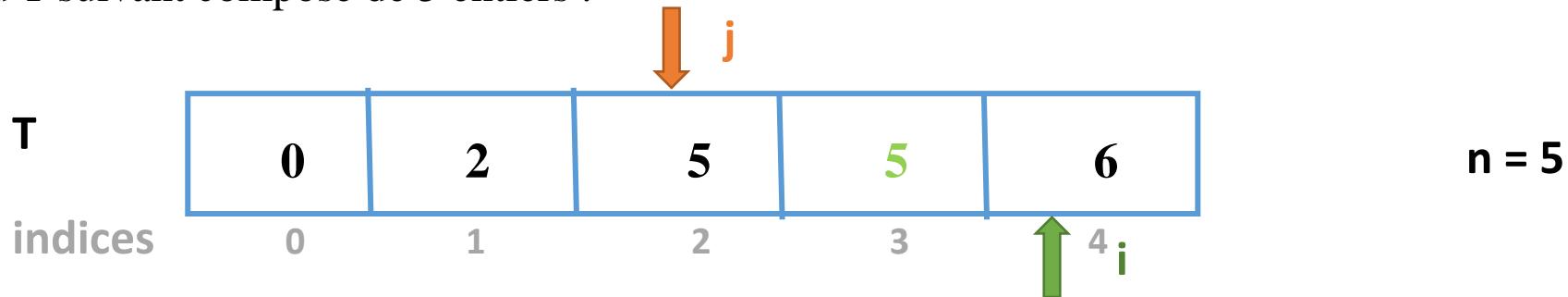
tmp
3

while ((3 > 0) && (5 > 3)) --> VRAI

{
 T[3] = T[2];
 j= 2;
}
T[j] = 3;

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3

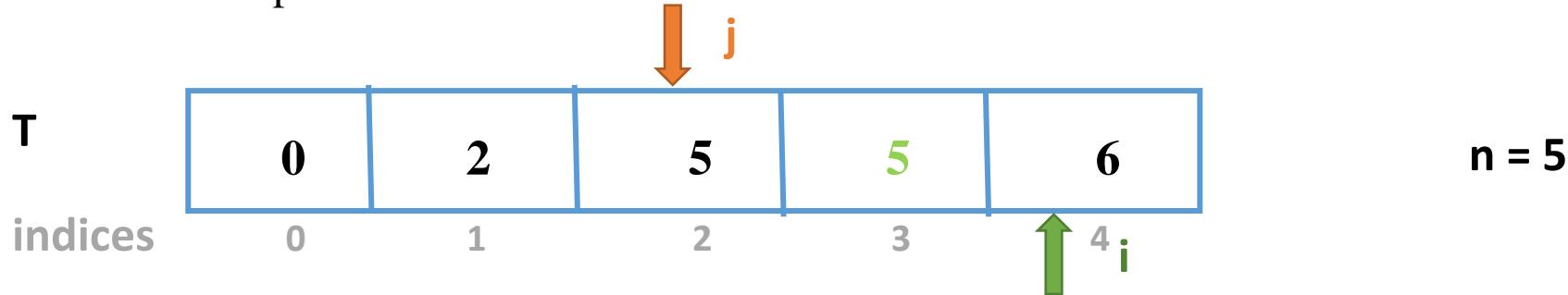
\rightarrow while ((3 > 0) && (5 > 3)) --> VRAI

{
 T[3] = T[2];
 j= 2;
}

$T[j] = 3;$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp ))
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}
```

tmp
3

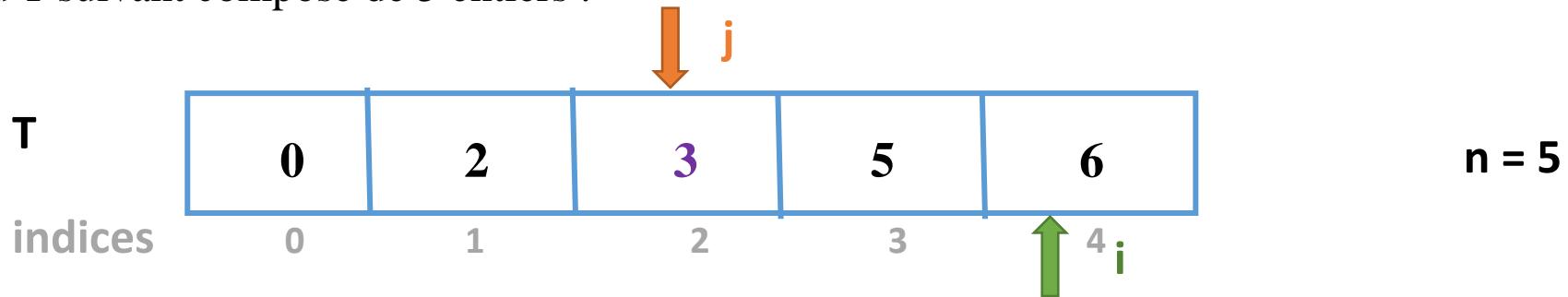
while ((2 > 0) && (2 > 3)) --> FAUX
i = 4

T[j] = 3;

A blue arrow points from the code line "while(j > 0 && (T[j-1] > tmp))" to the start of the while loop. A grey curved arrow indicates the loop's flow from the end of the loop body back to the condition.

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



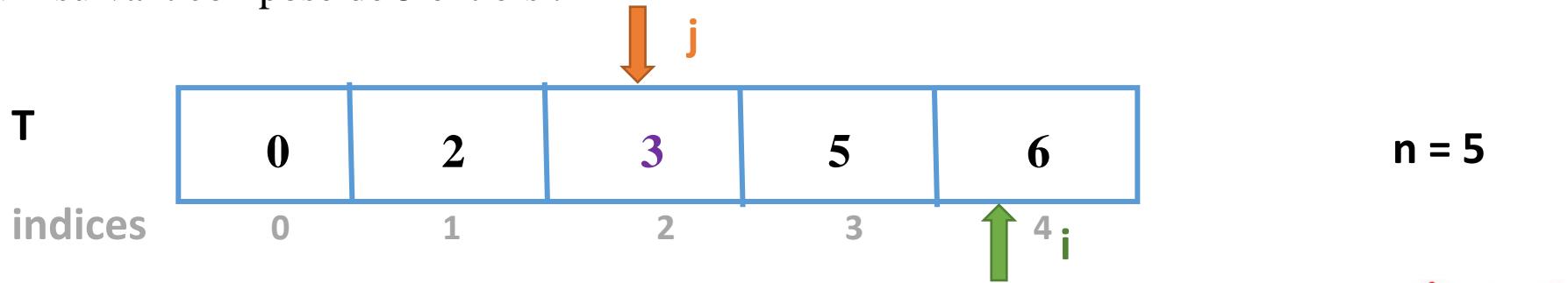
```
for (i=1;i<n;i++)
{
    tmp = T[i];
    j = i;
    while(j > 0 && (T[j-1] > tmp )) →
    {
        T[j] = T[j-1];
        j--;
    }
    T[j] = tmp;
}

tmp
3
while (( 2 > 0 ) && (2 > 3 ) ) --> FAUX
{
    T[2] = T[1];
    j= 1;
}
T[2] = 3;
```

$i = 4$

Exemple d'exécution

Soit le tableau T suivant composé de 5 entiers :



```
for (i=1;i<n;i++)  
{  
    tmp = T[i];  
    j = i;  
    while(j > 0 && (T[j-1] > tmp ))  
    {  
        T[j] = T[j-1];  
        j--;  
    }  
    T[j] = tmp;  
}
```

Le tableau est trié par ordre croissant