

Une illustration du fonctionnement de la fonction `partition` du quicksort en place, en particulier la boucle décrite au 1)a) de l'exercice 2, et les points `i` et `ii`.

On part de la liste $L = [4, 0, 6, 0, 4, 1, 7, 2, 2, 8]$. Le pivot sera donc 4, et on réordonne à l'aide d'une boucle `while` la liste $L[1:] = [0, 6, 0, 4, 1, 7, 2, 2, 8]$ pour que la première partie soit strictement inférieure au pivot, et la fin supérieure ou égale au pivot.

- `i` si $L[\text{debut}]$ est strictement plus petit que le pivot, $L[\text{debut}]$ est à sa place, et on augmente `debut` de 1.
- `ii` si $L[\text{debut}]$ est plus grand que le pivot, on le met dans les « plus grands », donc en fin de liste : on l'échange avec $L[\text{fin}-1]$, l'élément le plus à droite non encore trié, puis on diminue `fin` de 1.

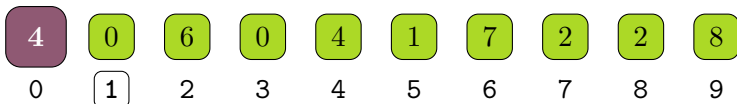
Légende :

- le pivot ;
- les éléments de $L[1:\text{debut}]$: la liste `plus_petits` en cours de remplissage ;
- les éléments de $L[\text{fin}:]$: la liste `plus_grands` en cours de remplissage ;
- l'élément $L[\text{debut}]$ en cours d'étude.
- 0 les éléments non encore étudiés.

On encadre les éléments d'indice `debut` et `fin`, et on entoure l'élément étudié (qui est toujours $L[\text{debut}]$).

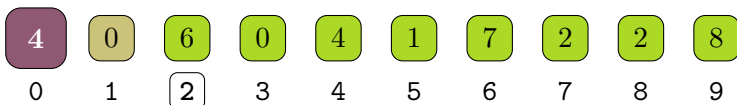
1) Initialisation :

`pivot = 4`
`debut = 1` $L[1:1] = []$ < pivot
`fin = 10` $L[10:] = []$ > pivot



2) Passage dans la boucle `while` : on est dans le cas `i`, donc rien ne bouge et `debut` augmente de 1.

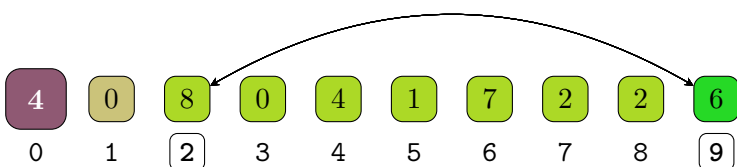
`debut = 2` $L[1:2] = [0]$ < pivot
`fin = 10` $L[10:] = []$ > pivot



3) Passage dans la boucle `while` : on est dans le cas `ii`, donc on échange $L[\text{debut}]$ avec $L[\text{fin}-1]$, l'élément le plus à droite non encore trié, puis on diminue `fin` de 1.

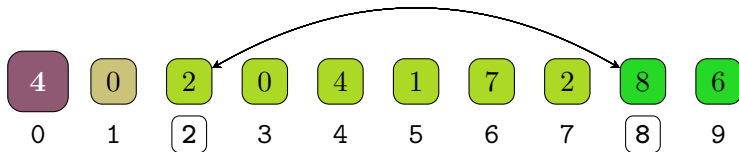
Remarque : le nouveau $L[\text{debut}]$ fait parti des éléments non trié, on ne connaît pas sa place par rapport au pivot.

`debut = 2` $L[1:2] = [0]$ < pivot
`fin = 9` $L[9:] = [6]$ > pivot



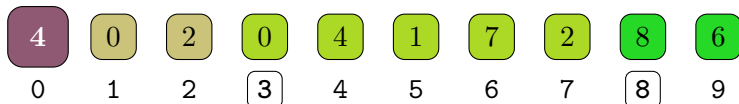
- 4) Passage dans la boucle **while** : on est dans le cas **ii**, donc échange et diminution de **fin** de 1.

debut = 2 L[1:2] = [0] < pivot
fin = 8 L[8:] = [8, 6] > pivot



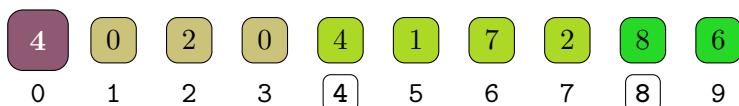
- 5) Passage dans la boucle **while** : on est dans le cas **i**, donc rien ne bouge et **debut** augmente de 1.

debut = 3 L[1:3] = [0, 2] < pivot
fin = 8 L[8:] = [8, 6] > pivot



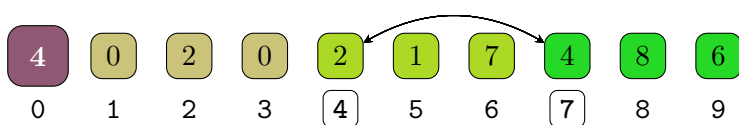
- 6) Passage dans la boucle **while** : on est dans le cas **i**, donc rien ne bouge et **debut** augmente de 1.

debut = 4 L[1:4] = [0, 2, 0] < pivot
fin = 8 L[8:] = [8, 6] > pivot



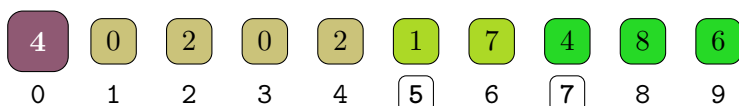
- 7) Passage dans la boucle **while** : on est dans le cas **ii**, donc échange et diminution de **fin** de 1.

debut = 4 L[1:4] = [0, 2, 0] < pivot
fin = 7 L[7:] = [4, 8, 6] > pivot



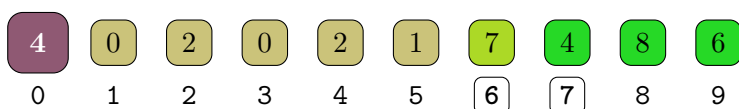
- 8) Passage dans la boucle **while** : on est dans le cas **i**, donc rien ne bouge et **debut** augmente de 1.

debut = 5 L[1:5] = [0, 2, 0, 2] < pivot
fin = 7 L[7:] = [4, 8, 6] > pivot



- 9) Passage dans la boucle **while** : on est dans le cas **i**, donc rien ne bouge et **debut** augmente de 1.

debut = 6 L[1:6] = [0, 2, 0, 2, 1] < pivot
fin = 7 L[7:] = [4, 8, 6] > pivot



- 10) Passage dans la boucle **while** : on est dans le cas **ii**, donc échange (avec lui-même, vu les indices) et diminution de **fin** de 1.

debut = 6 L[1:6] = [0, 2, 0, 2, 1] < pivot
fin = 6 L[6:] = [7, 4, 8, 6] > pivot

