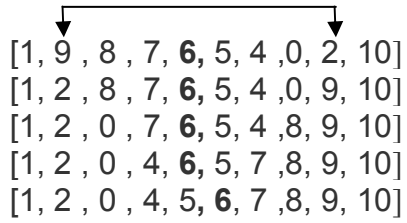
I.2 La récursivité pour le tri quicksort de Hoare

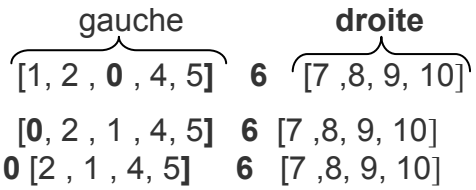
Le principe de l'algorithme du tri rapide ou tri de Hoare est :
1. On choisit un élément du tableau qui servira de pivot.
2. On effectue un **partitionnement** du tableau par permutation de valeurs dans le tableau de manière à ce que :
   a- Toutes les valeurs inférieures au pivot soient placées à sa gauche.
   b- Toutes les valeurs supérieures au pivot soient placées à sa droite.
3. On applique récursivement ce partitionnement sur les deux parties du tableau (à gauche et à droite du pivot).

Partitionnement :
Choisir une valeur de pivot
Pour toutes les valeurs du tableau à partitionner
  On parcourt de gauche à droite jusqu'à rencontrer un élément supérieur au pivot ou terminé.
  On parcourt de droite à gauche jusqu'à rencontrer un élément inférieur au pivot ou terminé.
  On permute ces deux éléments si non terminé.

Exemple :Soit [1, 9 , 8 , 7, **6,** 5, 4 ,0, 2, 10] le tableau à trier et la valeur 6 du milieu du tableau la valeur pivot.
Le premier partitionnement réalise les 4 permutations comme suit :



On obtient




```
int partition(int V[], int deb, int fin)
{  int pivot,i,j;
   pivot=V[(deb+fin)/2];    i=deb;  j=fin;
  while (i<j)
  {  while (pivot>V[i]) i++;
     while (pivot<V[j]) --j;
     if (i<j) swap(V,i,j); // si non terminé
  }
  return j;
}
void quickSort(int V[], int deb, int fin)
{  int ppivot;
   if (deb<fin) {  ppivot=partition(V,deb,fin);
                   quickSort(V,deb,ppivot-1);
                   quickSort(V,ppivot+1,fin);
                }
}
```