Tri rapide (Hoare 62)

```
function TrouverPivot(i,j : integer) : integer ;
var
  PremClé : TypeClé ;
  k : integer ;
begin
  PremClé := T[i] ;
  k := i + 1 ;
  while ((T[k] = PremClé) and (k < j)) do
    k := k + 1 ;
  if T[k] > PremClé then TrouverPivot := k
  else if T[k] < PremClé then TrouverPivot := i
       else TrouverPivot := 0 ;
end ;

function Partition(i,j : integer; pivot : TypeCle) : integer ;
var
  gauche, droite : integer ;
begin
  gauche := i ; droite := j ;
  repeat
    echanger(T[gauche], T[droite]) ;
    while (T[gauche] < pivot) do
      gauche := gauche + 1 ;
    while (T[droite] >= pivot) do
      droite := droite - 1 ;
  until gauche > droite ;
  Partition := gauche ;
end ;

Procedure TriRapide(i,j : integer) ;
var
  pivot : TypeCle ;
  PosPivot : integer ;
  k : integer ;
begin
  PosPivot := TrouverPivot(i,j) ;
  if PosPivot <> 0 then begin
    pivot := T[PosPivot] ;
    k := Partition(i,j,pivot) ;
    TriRapide(i, k-1) ;
    TriRapide(k,j) ;
  end ;
end ;
```