

Fonctions récursives

- 1- Voici un algorithme qui trouve l'élément maximal d'une liste A.

```
AlgoMax(A, n, Max)
Si n = taille de la liste
    Max := A [n-1]
Si n > 0
    Si Max < A [n-2]
        Max = A [n-2]
    AlgoMax(A, n-1, Max)
```

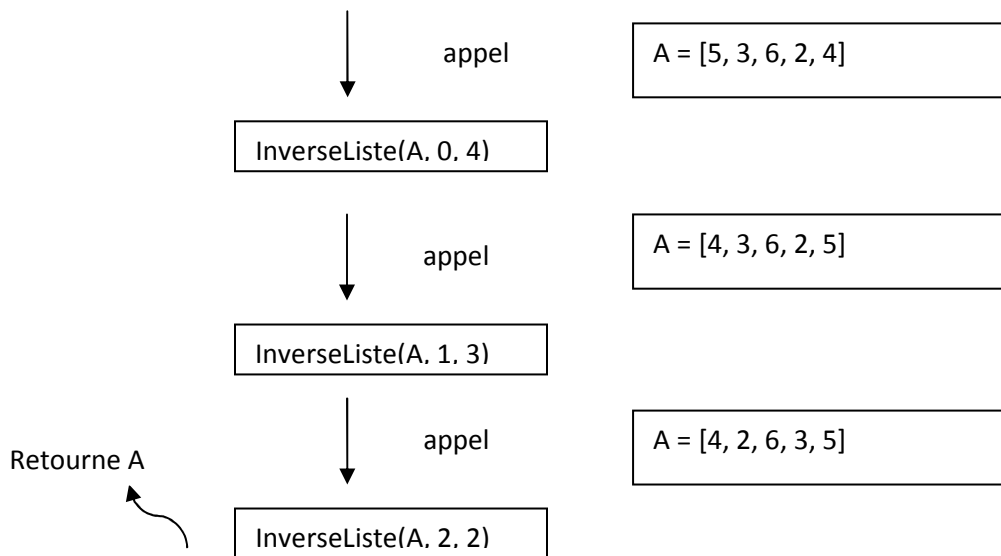
D'autres algorithmes sont bons pourvu qu'on parcourt la liste au complet, peu importe l'ordre de parcours. Ici, on n'est pas obligés de retourner Max puisqu'on y accède par paramètre. Les indices de la liste vont de 0 à n-1.

- 2- Algorithme InverseListe (A, i, j)

```
Si i > j alors
    Échanger A[i] et A[j]
    InverseListe(A, i+1, j-1)
```

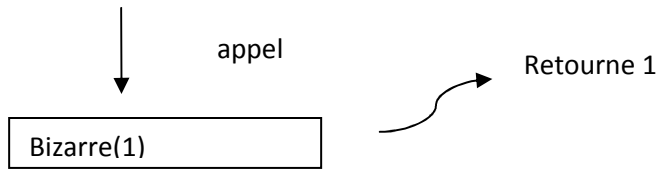
Retourner A

Trace de l'algorithme pour InverseListe(A, 0, 4) avec A = [5,3,6,2,4]

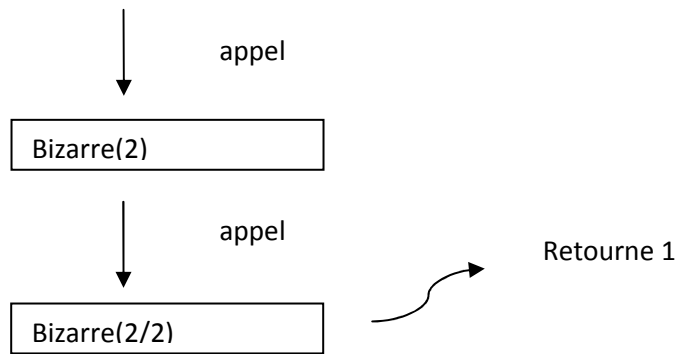


3- Algorithme bizarre qui retourne toujours 1

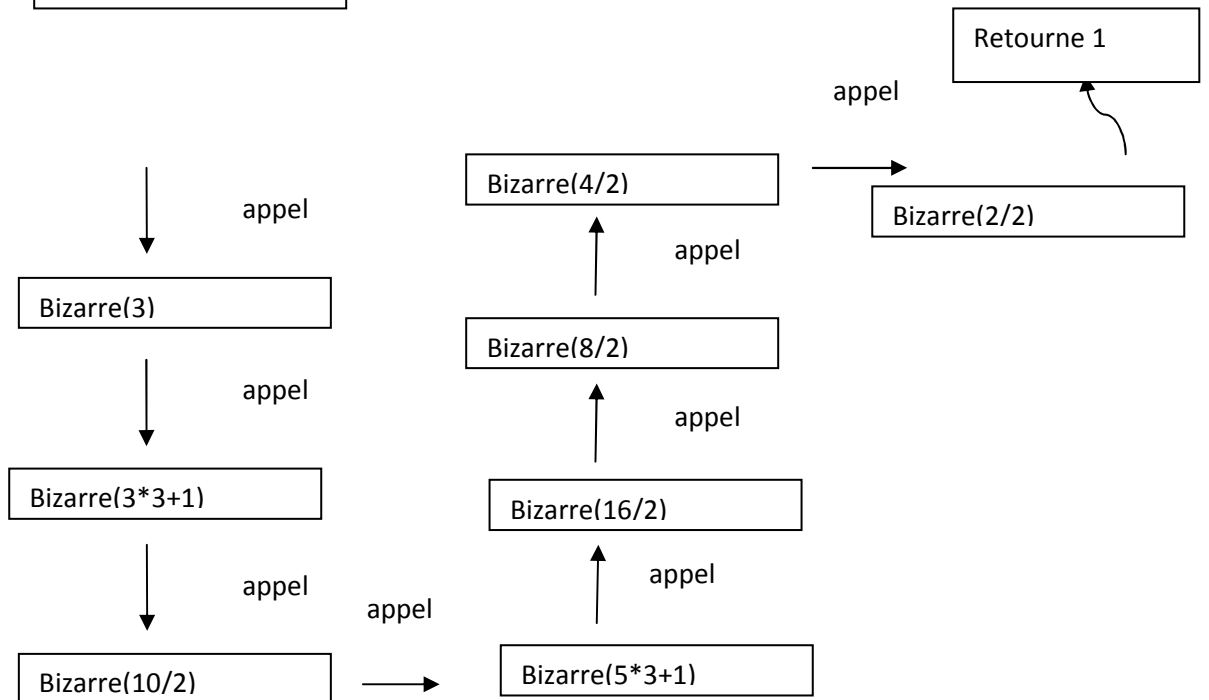
Pour n = 1



Pour n = 2



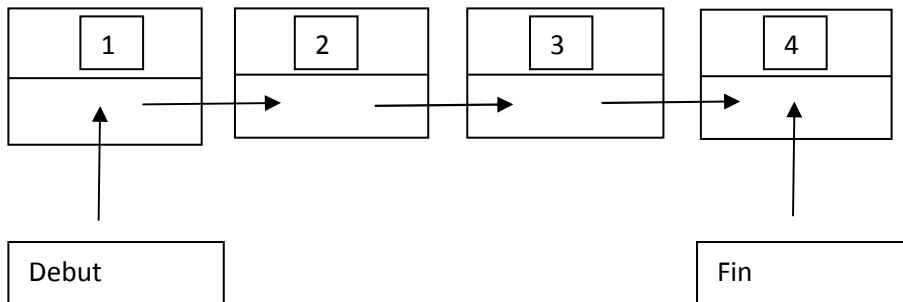
Pour n = 3



4. Après la longue analyse faite en démo, voici un algorithme pour inverser une liste chaînée. On fait seulement parcourir la liste une fois du début à la fin, en prenant les nœuds deux par deux et en inversant le sens de la flèche.

```
inverseListe(noeud, pere )  
  si (courant.suivant != vide)  
    inverseListe(courant.suivant, courant);  
  courant.suivant = pere;
```

LISTE



LISTE

INVERSÉE

