TP7 TNSI	Algorithmique – 2 – Méthode « Divide and Conquer »

L'objectif de ce TD est d'introduire une méthode de terminale pour résoudre des problèmes d'algorithmique. Il s'agit de la méthode « diviser pour régner ».

I - Diviser pour régner

A - Description

Il s'agit dans cette méthode

- de diviser un problème en plusieurs petits sous-problèmes indépendants et plus faciles que le problème global,
- puis, de résoudre ces petits problèmes,
- et enfin de fusionner les résultats des petits problèmes.

Le plus souvent, il s'agit d'une approche récursive ou chaque appel récursif permet la réduction de la taille du problème d'un facteur 2 ou plus (et pas seulement de le réduire de 1).

B - Exemple (déjà vu en terminale): Exponentiation rapide

1 - Constats

Notre fonction Expo a pour entrée a , flottant et n , entier positif et renvoie a^n . On remarque que $a^0=1$ et $a^n=a*a^{n-1}$ ce qui permet un algorithme récursif pour lequel n réduit de 1 à chaque appel.

On peut toutefois mettre en œuvre une méthode « diviser pour régner » en remarquant que :

- si n est pair, $a^n=(a*a)^{\frac{n}{2}}$
- si n est impair, $a^n=a*(a*a)^{\frac{n-1}{2}}$

On remarque qu'à chaque calcul on divise n par 2, ce qui rapproche d'autant plus vite n de 0, notre cas de base.

2 - Algorithme

```
Fonction Expo (a,n) :
     In :  a , flottant et  n , entier positif
     Out :   a^n
     Si n = 0,
          Renvoyer 1
     Si n est pair,
          Renvoyer Expo (a*a, n/2)
     Sinon,
          Renvoyer a*Expo (a*a, (n-1)/2)
```

3 - Questions

a - Décrire la pile d'appel lors du calcul de Expo(2,9).

b - Quelle est la complexité temporelle de cet algorithme ?

c - Prouver que cet algorithme se termine. (On pourrait aussi dire : « Prouver la terminaison de cet algorithme »).

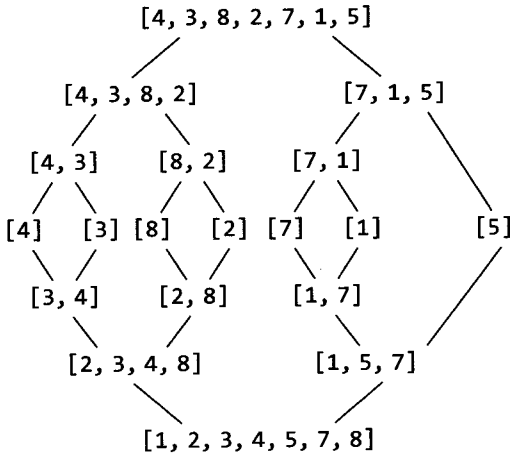
Cet algorithme a déjà été implémenté dans le TP « Récursivité ».

II - Algorithme de tri – le tri fusion

On revient ici sur les tris, à la recherche d'un algorithme plus efficace que les tris par insertion et par sélection. Le tri fusion s'appuie sur une méthode « Diviser pour régner ».

A - Tri fusion

1 - Exemple



2 - Description

TriFusion a pour entrée une liste de nombres. Et pour sortie la liste de nombres triée par ordre croissant. Les étapes de la méthode « Diviser pour régner » sont :

1. Découper la liste en sous-listes jusqu'à obtenir des listes de longueur 1.
2. Classer ces listes de longueur 1 (rien à faire ici).
3. Fusionner les listes classées deux à deux, comme elles sont classées, on n'a besoin que de comparer le premier élément de chaque liste à chaque étape.

3 - Algorithme

On part de la description du tri fusion pour obtenir l'algorithme :

```
fonction TriFusion (lst):
    In : une liste lst de nombres
    Out : les éléments de lst sont triés du plus petit au plus grand
    Si longueur(lst)==1 :
        renvoyer lst
    Sinon :
        découper lst en deux listes : lst1 et lst2
        renvoyer la fusion de TriFusion(lst1) et TriFusion(lst2)
```

Il nous reste donc à expliciter la fonction découper et la fonction fusion.

```
Fonction Fusion (lst1,lst2):
    In : deux listes triées lst1 et lst2
    Out : Une liste triée des éléments de lst1 et lst2
    Si lst1 est vide :
        renvoyer lst2
    Si lst2 est vide :
        renvoyer lst1
    Si lst1[0]<lst2[0] :
        renvoyer [lst1[0],Fusion(lst1[1:],lst2)]
    Si lst2[0]<lst1[0] :
        renvoyer [lst2[0],Fusion(lst2[1:],lst1)]
```

```
Fonction Decouper(lst) :
    In : une liste
    Out : deux listes comptant le même nombre d'éléments, à un près.
     n=longueur(lst)
    moitie=n//2 # division entière
    renvoyer  lst[0:moitie-1],lst[moitie :n-1]
```

4 - Implémentation

a - Programmer cet algorithme en python dans le fichier ImplementationComplexite (fonction TriFusion).

b - Tests

Tester votre algorithme avec le jeu de test du TD précédent.

5 - Complexité

a - Complexité temporelle

Combien d'opérations sont nécessaires dans le pire des cas pour trier une liste de longueur n avec le tri fusion ?

b - Grâce à la fonction trace, vérifier votre calcul du a avec python.

6 - Complexité spatiale

Quel est l'espace occupé par ce tri ?
En déduire sa complexité spatiale.