

Devoir Maison n°3 d'Informatique

pour Mercredi 29 Janvier 2025

Exercice 1 : méthode d'exponentiation rapide

Soient $a \in \mathbb{R}$ et $n \in \mathbb{N}$. Pour calculer a^n , on peut multiplier a par lui-même n fois. L'algorithme ci-contre permet d'accélérer ce calcul ; quand il se termine, r vaut a^n et Nbre est le nombre de boucles effectuées (c'est donc un indicateur de complexité).

```

r ← 1
Nbre ← 0
TANT QUE n>0
  Nbre ← Nbre+1
  SI n est pair
  ALORS
    a ← a²
    n ← n/2
  SINON
    r ← r*a
    n ← n-1
FIN du TANT QUE
Renvoyer (r,Nbre)

```

- 1) Utiliser cet algorithme pour créer en Python une fonction `Puissance(a,n)` renvoyant le couple (a^n, Nbre) .

Exemple : `Puissance(3,8)` doit renvoyer $(6561, 4)$.

- 2) En modifiant légèrement le code de cette fonction, créer une fonction `Boucles(n)` qui renvoie le nombre de boucles nécessaires au calcul de 2^n .

Exemple : `Boucles(13)` doit renvoyer 6.

- 3) On veut maintenant trouver le lien entre le paramètre n et `Boucles(n)`.
 - (a) Représenter `Boucles(n)` en fonction de n pour $n=10, 20, 30, \dots, 1000$.
 - (b) Représenter `Boucles(10**n)` en fonction de n pour $n=1, 2, 3, \dots, 10$.
 - (c) Représenter `Boucles(n)` en fonction de $\log(n)$ pour $n=10, 20, 30, \dots, 1000$.
 - (d) Proposer alors une estimation « à la louche » de `Boucles(n)` en fonction de n .
Quelle est donc la complexité de l'algorithme ?

Exercice 2 : tri d'une liste

Le but de cet exercice est de créer une fonction `Tri(L)` qui prend les éléments d'une liste de nombres L et les range, dans l'ordre croissant, dans une nouvelle liste.

- 1) Commencer par créer une fonction `mini(L)` qui renvoie le minimum d'une liste L .

Nota bene : nous avons déjà croisé cette fonction et l'algorithme associé à plusieurs reprises ...

Exemple : `mini([1,6,5,-3,8,2])` doit renvoyer -3.

- 2) Voici comment procéder pour trier une liste L :

- ❶ on crée une liste vide $L1=[]$;
- ❷ on détermine le minimum de L , on l'ajoute à $L1$ et on le supprime de L ;
- ❸ on répète l'étape précédente jusqu'à épuisement de la liste L ;
- ❹ la liste $L1$ contient alors les éléments de L rangés dans l'ordre croissant.

Utiliser cet algorithme pour programmer la fonction `Tri(L)`.

Nota bene : elle pourra utiliser la fonction `mini(L)` créée précédemment ...

Exemple : `Tri([1,6,5,-3,8,2])` doit renvoyer $[-3, 1, 2, 5, 6, 8]$.