Exponentiation rapide

On considère l'algorithme suivant :

Algorithme 1 : Exp

```
   Input : (x, n)
1  if n = 0 then
       Output : 1
2  if n = 1 then
       Output : x
3  if n est pair then
4      y ← Exp(x, n/2)
       Output : y * y
5  else
6      y ← Exp(x, n/2) // Division entière
       Output : y * y * x
```

1. Montrez que quelque soient les entiers positifs x et n, l'algorithme Exp renvoie la valeur de x^n.

2. Donnez une borne sur le nombre de multiplications effectué par l'algorithme.

3. L'exponentiation modulo n est-il un problème "facile" ou "difficile" ?

RSA est-il vraiment NP-difficile ?

1. Écrire un algorithme prenant en entrée un entier n et renvoyant un diviseur premier de n en testant successivement tous les diviseurs possibles.

2. Quel est la complexité de l'algorithme ?

3. Peut-on utiliser cet algorithme pour résoudre le problème RSA ? Que concluez-vous ?

Algorithme de tri

Trier un ensemble d'objet n'a pas forcément une application immédiate mais il est souvent plus facile de travailler avec des structures ordonnées, par exemple pour pouvoir effectuer rapidement des recherches. L'un des algorithmes les plus connus pour cela est le tri fusion qui est un algorithme récursif.

1. Écrire un algorithme (fusion) qui prend en entrée deux listes triées et renvoie en sortie une liste triée comportant les éléments des deux listes.
2. Écrire un algorithme qui prend en entrée une liste L de taille n (non triée) et renvoie en sortie deux listes L_1 et L_2 de taille $n/2$ tel que $L = L_1 \cup L_2$.
3. En utilisant des deux procédures précédente écrire un algorithme récursif de tri de liste. Analyser sa complexité.