TD "Complexité & Preuves de Programmes"

1. Addition

Donnez deux algorithmes qui additionnent deux nombres entiers.

- Un premier qui est celui utilisé quand on fait une addition "à la main" Indications
- Un deuxième qui utilise d'autres "fonctions de base" Indications

Comparez les complexités de ces algorithmes. Solution (important)

2. Puissance

Donnez deux algorithmes, un itératif (Solution) & un récursif (Solution), rendant, à partir de deux entiers positifs x & y, le nombre x^y.

Donnez leurs complexités. Solution

Prouvez les. Solution

3. Maximum d'un tableau

Donnez deux algorithmes, un itératif (Solution) & un récursif (Solution), qui calculent la plus grande valeur d'un tableau de nombres.

Comparez leurs fonctionnements. Solution

4. Algorithme Mystère

On considère l'algorithme suivant:

def $misterioso(A, B)$:
$\;\;\;\;$ # A & B sont des entiers positifs
$\;\;\;\;$ $x \gets A ;\, y \gets B ;\, r \gets 1$
$\;\;\;\;$ # Ligne 1
$\;\;\;\;$ while $y \neq 0$:
$\;\;\;\;$ $\;\;\;\;$ # Ligne 2
$\;\;\;\;$ $\;\;\;\;$ if y impair:
$\;\;\;\;$ $\;\;\;\;$ $\;\;\;\;$ $r \gets r \times x$
$\;\;\;\;$ $\;\;\;\;$ $\;\;\;\;$ $y \gets y -1$
$\;\;\;\;$ $\;\;\;\;$ else:
$\;\;\;\;$ $\;\;\;\;$ $\;\;\;\;$ $x \gets x\times x$
$\;\;\;\;$ $\;\;\;\;$ $\;\;\;\;$ $y \gets y /2$
$\;\;\;\;$ $\;\;\;\;$ # Ligne 3
$\;\;\;\;$ # Ligne 4

$\;\;\;\;$ return r

Que fait cet algorithme (indication) ? Justifiez votre réponse. Réponse

Quelle sa complexité dans le cas le meilleur (indication), le pire (indication), en moyenne ?

5. Tris Naïfs

Prouvez les algorithmes de tris simples vus en cours (tri par sélection & tri par insertion)

Indications

6. Palindromes

Implémentez un algorithme récursif permettant de savoir si un tableau de caractères est un palindrome (un palindrome se lit "à l'endroit" & "à l'envers" de la même façon, comme par exemple "À l'étape, épate-la !" (on considère ni la ponctuation, ni les espaces, ni les accents)). Quelle est sa complexité?

Solution

7. Tours de Hanoï

Les *tours de Hanoï* sont un célèbre casse tête qui consiste à déplacer n disques de diamètres différents d'une tour de "départ" à une tour d' "arrivée" en passant par une tour "intermédiaire", tout en respectant les règles suivantes :

- on ne peut déplacer qu'un disque à la fois,
- on ne peut placer un disque sur un disque plus petit que lui.

On suppose que cette dernière règle est également respectée dans la configuration de départ.

Donnez un algorithme récursif permettant de résoudre le problème. Quelle est sa complexité? Peut-on faire mieux?

Indication