

ED3 (TD) - Un peu d'algorithmique

Objectifs

Concevoir des algorithmes simples en pseudo-code, savoir évaluer sa solution. Savoir reconnaître une suite récurrente simple dans un problème donné.

1 Petits exercices

EXERCICE 1 Soient a et b deux entiers positifs. Écrire un algorithme itératif permettant de calculer $a \times b$ en utilisant l'addition. Combien coûte votre algorithme en termes d'additions ?

EXERCICE 2 Écrire un algorithme itératif permettant de calculer le quotient et le reste de la division euclidienne de a par b sans utiliser la division ni le modulo. Si $a = 37$ et $b = 5$ votre algorithme devra calculer $q = 7$ et $r = 2$.

2 Un problème de complexité algorithmique

Fonctions préparatoires Écrire les fonctions suivantes en pseudo-code (algorithmes itératifs) :

- Puissance : Fonction `puiss(x,n)`: Réel avec x réel et n entier supposé positif ou nul.
- Factorielle : Fonction `fact(n)`: Entier avec n entier supposé positif ou nul.

Pour chacune de ces fonctions, on donnera le coût en nombre de multiplications en fonction de n .

Allons-y! On rappelle le DL en 0 de e^x :

$$E_n = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

On va utiliser ce DL pour calculer des valeurs approchées de e^x avec x petit (réel, proche de 0).

1. Étant donné un réel x et un entier k , écrire un algorithme permettant de calculer $E_k(x)$ valeur approchée de e^x .

Indication : Comment calculer E_i si l'on connaît E_{i-1} ?.

On utilisera obligatoirement les fonctions écrites précédemment.

2. Combien coûte le calcul de E_n ?
3. Remplacer les appels de fonctions par un calcul de proche en proche de factorielle et de puissance, puis évaluer le coût.
4. Écrire un programme principal demandant un flottant x et un entier k à l'utilisateur, appelant l'algorithme précédent et imprimant E_k une valeur approchée de e^x .