

RICM1 - 2004/05
Langage et Programmation 2

TD7 : Ordre d'évaluation et mesures de performance

Exercice 1

Écrire une fonction f qui calcule $5! + n$ de telle sorte que le calcul de $5!$ est effectué une seule fois même si f est appelée plusieurs fois.

Exercice 2

Écrire une fonction qui permet d'exécuter un calcul n fois.

Exercice 3

Retrouver la fonction *duree* qui, appliquée à une fonction f et à un argument x , donne le temps de calcul de $f\ x$.

Peut-on l'utiliser sur une fonction à deux arguments ?

Exercice 4

Dans cet exercice, on considère un type structure non spécifié (qui pourrait être, par exemple, une liste ou un arbre binaire).

Exercice 4-1

On se donne :

- une fonction *mk_str* qui, pour un entier donné n , rend une structure de taille n
- une taille initiale *ini*
- une fonction de progression *prog* qui, à partir d'une taille, donne la taille suivante
- une longueur *nb*

Écrire une fonction *mk_echantillon* qui rend une liste constituée de nb couples dont le premier élément est une taille t et le deuxième est une structure de taille t , avec t valant successivement *ini*, *prog ini*, ... Faire en sorte que dans *mk_echantillon*, l'évaluation de *mk_str* soit effectuée en premier sur les structures de taille les plus petites.

Exercice 4-2

On se donne :

- une liste de fonctions *lf* dont on veut mesurer les performances, s'appliquant toutes à des structures
- une liste *echantillon* de couples (t , structure de taille t)
- un nombre de répétitions *nb_rep*

Écrire une fonction *mesures* qui, pour chaque fonction f de *lf*, parcourt *echantillon* en évaluant *nb_rep* fois f sur chaque structure et affiche la durée d'exécution.