Maple

TP n°4 : La récursivité

Les procédures que nous avons vues jusqu'alors étaient des procédures itératives. Elles sont le plus souvent rapides mais pas nécessairement faciles à mettre en oeuvre. Une autre méthode consiste à employer la récursivité : Une procédure pouvant appelée d'autres procédures, rien ne l'empêche de s'appeler elle-même. On parle alors de récursion ou de procédure récursive.

Un exemple simple et classique est la fonction factorielle. En effet, $n! = \text{Fact}(n) = n \times \text{Fact}(n-1)$. Comme on peut le remarquer dans cet exemple, un point important est de savoir quand on doit s'arrêter. Ici, la condition de sortie du programme est $n = 0$. Dans ce cas, la fonction doit renvoyer $\text{Fact}(0) = 1$.

Cela donne donc :

```
factorielle:=proc(n::nonneg)
if n=0 then
     1
else
     n * factorielle (n-1)
fi;
end;
```

Les procédures récursives sont particulièrement bien adaptées au niveau de l'écriture de la procédure pour toutes les suites définies par une récurrence du type $u_{n+1} = f(u_n)$.

Exercice 1

Soit p un entier non nul.
Donner une procédure récursive qui permette de déterminer la valeur du terme d'ordre p de la suite numérique u dite de Fibonnacci définie par $u_0 = u_1 = 1$ et $u_{n+2} = u_{n+1} + u_n$.
Info : la complexité de ce calcul est $O\left(\frac{1+\sqrt{5}}{2}\right)^p$.

Exercice 2

Soit la suite numérique u définie par $u_0 = 5$ et $u_{n+1} = \frac{1}{5}u_n + 1 + \frac{1}{5^n}$.

1. Ecrire une procédure récursive donnant la valeur "exacte" de u_p pour un entier p donné.
2. Ecrire une procédure récursive donnant la valeur approchée de u_p pour un entier p donné.
3. Utilisez les 2 procédures pour calculer u_{10} et u_{20} avec des décimaux représentatifs.

Exercice 3

Soient les suites u et v définies par
$$u_0 = 1,$$
$$v_0 = 2\ ,$$
$$u_{n+1} = \frac{1}{5}u_n - \frac{1}{2}v_n + 1 \text{ et}$$
$$v_{n+1} = \frac{1}{4}u_n + \frac{1}{3}v_n - 2.$$

1. Ecrire une procédure itérative donnant à la fois la valeur "exacte" et la valeur approchée de u_p et de v_p à 10^{-15} près pour un entier p donné.
2. Ecrire une procédure récursive donnant à la fois la valeur "exacte" et la valeur approchée de u_p et de v_p à 10^{-15} près pour un entier p donné.
3. Vérifier chacune des fonctions pour $p = 10, 20$.
4. Ajouter l'option `remember` à chacune des procédures dans le cas récursif. Vider la mémoire et refaire les calculs pour $p = 10, 20, 30$.

Exercice 4

1. Ecrire une procédure itérative (que nous appellerons `som_chiffr_it`) qui étant donné un entier renvoie la somme de ses chiffres.
2. Ecrire une procédure récursive (`som_chiffr_re`) qui fait la même chose.
3. Déterminer:
```
som_chiffr_it(som_chiffr_it(som_chiffr_it(4444^4444)));
som_chiffr_re(som_chiffr_re(som_chiffr_re(110^310)));
```

Exercice 5

Donner une procédure récursive répondant au problème du pgcd de l'exercice 4 du TP2.

Exercice 6

1. Donner une procédure récursive répondant au problème de la recherche par dichotomie de la solution de $f(x) = 0$ de l'exercice 9 du TP2.
   On supposera que les réels a et b vérifient bien $f(a)f(b) < 0$.
2. Tester cette procédure avec $f(x) = x^5 + 6x^3 - x^2 + x - 1$, $a = 0$ et $b = 1$.

Exercice 7

Soit $f : \mathbb{N}^* \to \mathbb{N}^*$ définie par $f(1) = 1$, $f(n) = n / 2$ si n pair et $f(n) = 3n + 1$ si $n \neq 1$ est impair.

1. Ecrire une procédure `syr` qui corresponde à f.
2. En utilisant la commande `seq`, afficher la suite `syr(n)` pour `n` variant de 0 à 30.
3. Ecrire une fonction itérative `syracuse_it(n,u0)` qui calcule le terme d'ordre n de la suite u définie par u_0 et par $u_{n+1} = f(u_n)$.
4. Ecrire une procédure récursive qui fasse la même chose.
5. a. Pour u_0 qui varie de 1 à 100, calculer u_{100}. On pourra utiliser la commande `seq`.
   b. Même question pour u_{1000} et u_0 qui varie de 1 à 100.
   c. Même question pour u_{1000} et u_0 qui varie de 1 à 1000.
   d. Conjecture? Preuve?

Exercice 8

Soient k et n deux entiers non nuls. Calculer $S(n,k) = \sum_{1 \le i_1 \le i_2 \le \ldots \le i_k \le n} \frac{1}{i_1 \times i_2 \times \ldots \times i_k}$.