TD 2
Récursivité

Exercice 1 : Programmation de formules mathématiques

Nous souhaitons calculer récursivement les fonctions suivantes :

- la factorielle $n!$;
- le calcul de r^n où r est un réel et n un entier ;
- le terme f_n de la suite de Fibonacci définie par :

$$f_n = \begin{cases} 0 \text{ si } n = 0 ; \\ 1 \text{ si } n = 1 ; \\ f_{n-1} + f_{n-2} \text{ sinon.} \end{cases} \tag{1}$$

- le terme u_n de la suite définie par :

$$u_n = \begin{cases} 3 \text{ si } n = 0 ; \\ 5 \text{ si } n = 1 ; \\ u_{n-1} \times u_{n-1} \times u_{n-2} \text{ sinon.} \end{cases} \tag{2}$$

- le terme v_n de la suite définie par :

$$v_n = \begin{cases} 2 \times v_{n-2} \text{ si } n \geq 3 \text{ et } n \text{ impair ;} \\ 1 \text{ sinon.} \end{cases} \tag{3}$$

- le terme w_n de la suite définie par :

$$w_n = \begin{cases} 1 \text{ si } n = 0 \text{ ou } n = 1 ; \\ \sum_{i=1}^{n-1} i \times w_i \text{ sinon.} \end{cases} \tag{4}$$

1) (TD) Donner le pseudo-code des fonctions qui permettent de calculer les formules mathématiques précédentes.

2) (TD) Donner leur complexité.

3) (TP) Les implémenter en Java.

Exercice 2 : Arbres binaires

Un *arbre binaire* est un arbre enraciné plongé dans le plan où chaque sommet possède 0 ou 2 fils. Un *noeud* est un sommet possédant 2 fils et une *feuille* est un sommet sans fils. Dans nos figures, les noeuds sont symbolisés par des ∘ et les feuilles par des •. La Figure **??** donne trois exemples d'arbres binaires.

1) (TD) Dessiner les arbres binaires formés de 0, 1, 2 puis 3 noeuds.

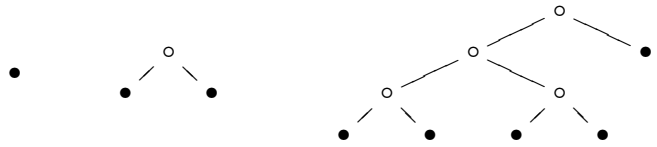

Figure 1: Quelques exemples d'arbres binaires

2) (TD) Donner le pseudo-code de la fonction qui à un entier n associe le nombre d'arbre binaires à n noeuds.

3) (TD) Donner la complexité de cette fonction.

4) (TP) Les implémenter en Java.

Exercice 3 : Le problème des dames

L'objectif du problème est de placer huit dames d'un jeu d'échecs sur un échiquier de 8x8 cases sans que les dames ne puissent se menacer mutuellement, conformément aux règles du jeu d'échecs (la couleur des pièces étant ignorée). Par conséquent, deux dames ne doivent jamais partager ni la même rangée, ni la même colonne, ni la même diagonale.

1) (TD) Dessiner quelques exemples de configurations acceptables, d'abord avec 6 dames, puis 7, et enfin 8.

2) (TD) Donner le pseudo-code de la fonction qui affiche toutes les solutions possible au problème des huit dames.

3) (TD) Donner la complexité de cette fonction.

4) (TP) Les implémenter en Java.

Exercice 4 : Tours de Hanoï

Le problème des tours de Hanoï est un jeu de réflexion imaginé par le mathématicien français Édouard Lucas qui consiste à déplacer des disques de différents diamètres d'une tour de *départ* à une tour d'*arrivée* en passant par une tour *intermédiaire* et ceci en un minimum de coups, tout en respectant les règles suivantes :

- un coup consiste à supprimer un disque qui se situe au sommet d'une tour et à le placer au sommet d'une autre tour ;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou dans une préalablement tour vide.

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

1) (TD) Donner les déplacements nécessaires pour résoudre le problème avec 1, 2 et 3 disques.

2) (TD) Donner le pseudo-code qui affiche les déplacement à effectuer pour résoudre le problèmes des tours de Hanoï.

3) (TD) Donner la complexité de cette fonction.

4) (TP) Les implémenter en Java.