Durée : 3 heures

Strutures algorithmiques autorisées

Dans tout ce sujet, lorsqu'il faudra rédiger un algorithme, vous pourrez le faire en pseudo-code avec, en plus de s'autoriser les opérations que vous jugerez être élémentaire, le droit de faire appel aux structures algorithmiques suivantes :

```
Pour x ∈ E faire
   instructions
FinPour
```

```
Tant que condition faire
   instructions
FinTantque
```

```
Si condition Alors
   instructions1
Sinon
   instructions2
```

```
Selon variables Avec
   |cas 1 -> instructions 1
   .
   .
   .
   |cas p -> instructions p
FinSelon
```

Avec E un ensemble quelconque dépendant de l'algorithme, `condition` une expression s'évaluant en un booléen (Vrai ou Faux), `instructions` une suite d'instructions quelconques,`variables` une ou plusieurs variables données sous la forme d'un n-uplet (possiblement $n = 1$) dépendant de l'algorithme et enfin `cas` une valeur possible pour le n-uplet `variables`.

Exercice 1

rappel On dit que $(E, >)$ est bien ordonné lorsque toute partie non vide F de E admet un minimum a. c'est à dire :

$$\forall x \in F, \quad x \geqslant a$$

On dit que $(E, >)$ est bien fondé lorsque toute partie non vide F de E admet au moins un élément minimal a. C'est à dire :

$$\forall x \in F, \quad a \geqslant x \Rightarrow a = x$$

1. Montrer qu'un ensemble bien ordonné est bien fondé.
2. Donner un exemple d'ensemble bien fondé mais non bien ordonné.
3. Étant donnés deux ordres bien fondés $(A, >_A)$ et $(B, >_B)$, en justifiant rapidement vos réponses, indiquer lesquels parmi les suivants sont aussi des ordres bien fondés :
   (a) $(A \times B, >_{A\times B})$ avec $(a, b) >_{A\times B} (a', b')$ si et seulement si $a >_A a'$ et $b >_B b'$.
   (b) $(A \times B, >_{A\times' B})$ avec $(a, b) >_{A\times' B} (a', b')$ si et seulement si ($a >_A a'$ et $b = b'$) ou ($a = a'$ et $b >_B b'$).
   (c) $(A \cup B, >_{A\cup B})$ avec $e >_{A\cup' B} e'$ si et seulement si ($e \in A$ et $e' \in A$ et $e >_A e'$) ou ($e \in B$ et $e' \in B$ et $e >_B e'$).

Exercice 2

On définit par induction l'ensemble des arbres binaires sur les entiers par :

- Pour tout entier `n`, l'élément `Feuille(n)` est un arbre binaire sur les entiers que l'on qualifie de feuille.
- Si `a` est un arbre binaire sur les entiers alors l'élément `N_unaire(a)` est un arbre binaire sur les entier.
- Si `ag` et `ad` sont des arbres binaires sur les entiers alors l'élément `N_bin(ag,ad)` est un arbre binaire sur les entier.

Dans la construction d'un arbre binaire sur les entiers quelconques, on qualifiera chaque occurence des constructeurs `N_unaire` et `N_bin` de nœud interne de l'arbre.

1. Dessiner tous les arbres binaires ayant exactement trois nœuds internes. (On ne représentera pas les feuilles et les entiers qui y sont attachés.)
2. Définir en pseudo-code un algorithme récursif prenant en entrée un arbre binaire sur les entiers `a` et renvoyant le nombre de feuilles de ce dernier.
3. Définir en pseudo-code un algorithme récursif prenant en entrée un arbre binaire sur les entiers `a` et renvoyant le nombre de nœuds internes de ce dernier.
4. Définir en pseudo-code un algorithme récursif prenant en entrée un arbre binaire sur les entiers `a` ainsi qu'un entier `n` et qui renvoie le booléen Vrai si l'entier `n` apparaît parmi les feuilles de `a` et le booléen Faux sinon.
5. Définir en pseudo-code un algorithme récursif prenant en entrée un arbre binaire sur les entiers `a` et renvoyant la somme de tous les entiers apparaissant dans ses feuilles.
6. Donner un ensemble de règles de construction qui permettent de construire l'ensemble des arbres dont les nœuds internes ont 1, 2 ou 3 successeurs.

Exercice 3

Les universités anglophones nord-américaines sont célèbres pour leurs associations étudiantes, leurs fraternités, ou leurs sororités. Celles-ci développent souvent une culture du secret.

Vous venez de rejoindre la fraternité $\Gamma\alpha\chi$. Lors de votre initiation, vous avez appris les règles gouvernant les discussions pour que seuls les membres puissent en comprendre le contenu. Lorsque plusieurs membres prennent la parole sur un sujet donné, soit ils disent tous la vérité, soit ils mentent tous. Si quelqu'un intervient sur plusieurs sujets, il peut avoir un rôle différent pour chaque sujet.

Pour être définitivement admis, vous devez démontrer votre maîtrise de ces règles. Vous participez à une discussion avec trois membres de la fraternité que nous appellerons A, B et C. Ceux-ci vous indiquent comment rejoindre la salle d'intronisation. Si vous n'arrivez pas à rejoindre directement cette salle, vous ne serez pas admis.

Un premier sujet est abordé concernant la pièce dans laquelle se tiendra la cérémonie :

A : «La cérémonie se tiendra dans le gymnase »

C : «Non, elle ne se tiendra pas dans cette pièce »

A : «Ou alors dans le réfectoire »

Nous noterons G et R les variables propositionnelles associées à la pièce où se tiendra la cérémonie. Nous noterons A_1 et C_1 les formules propositionnelles correspondant aux affirmations de A et C sur le premier sujet.

Puis, un second sujet est traité concernant les escaliers qui permettent de rejoindre la pièce où se tiendra la cérémonie.

A : «Le premier escalier conduit à l'intronisation »

C : «Tu as raison »

A : «Le troisième escalier y conduit aussi »

B : «Si le deuxième escalier y conduit, alors le troisième n'y conduit pas »

C : «Le deuxième n'y conduit pas »

Nous noterons E_1, E_2, E_3 les variables propositionnelles correspondant au fait que le premier, le deuxième, le troisième escalier conduisent à la salle de cérémonie.

Nous noterons A_2, B_2 et C_2 les formules propositionnelles correspondant aux affirmations de A, B et C sur le second sujet.

1. Représenter le premier sujet abordé sous la forme d'une formule du calcul des propositions dépendant des formules A_1, et C_1.
2. Représenter les informations données par les participants sous la forme de deux formules du calcul des propositions A_1 et C_1 dépendant des variables G et R.
3. En utilisant le calcul des propositions (résolution avec les formules de De Morgan), déterminer dans quelle pièce vous devez vous rendre pour rejoindre la cérémonie.
4. Représenter le second sujet abordé sous la forme d'une formule du calcul des propositions dépendant des formules A_2, B_2 et C_2.

5. Représenter les informations données par les participants sous la forme de trois formules du calcul des propositions A_2, B_2 et C_2 dépendant des variables E_1, E_2 et E_3.
6. En utilisant le calcul des propositions (résolution avec les tables de vérité), déterminer l'escalier que vous devez suivre pour rejoindre la cérémonie.
7. En admettant que les trois participants aient menti, pouviez vous prendre d'autres escaliers ? Si oui, le ou lesquels ?

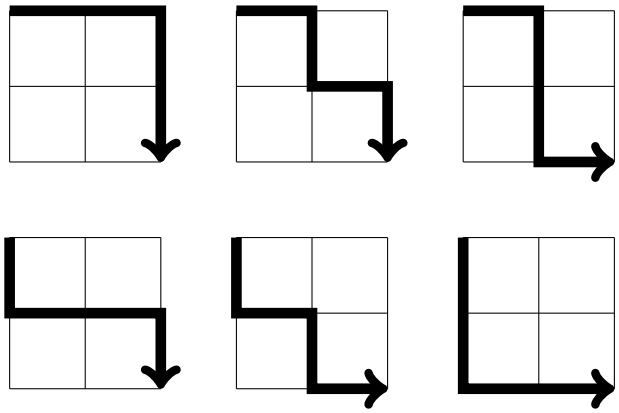
Exercice 4

On rappel que les listes chaînées d'éléments de type `'a` sont définies comme suit :
- Quelque soit le type `'a`, l'élément `[]` est une liste chaînée d'éléments de type `'a`.
- Si `x` est un élément de type `'a` et `q` est une liste chaînée d'éléments de type `'a` alors `x::q` est une liste chaînée d'éléments de type `'a`.

1. Ecrire en pseudo-code une fonction récursive `insertion x l` de type `'a -> 'a list -> 'a list` qui à un élément `x` et une liste chaînée `l` d'éléments du même type triée dans l'ordre croissant renvoie une liste contenant les éléments de `l` avec en plus `x` et toujours triée dans l'ordre croissant.
2. Ecrire une fonction récursive `tri_insertion l` de type `'a list -> 'a list` qui à une liste `l` renvoie une liste composée des même éléments mais triée dans l'ordre croissant. Cette fonction devra appeler la fonction précédente.
3. Donner la complexité de `tri_insertion l` dans le pire cas et le meilleur cas en fonction de la taille de la liste `l`. Justifier votre réponse.

Exercice 5

On considère un carré de taille $n \in \mathbb{N}^*$ quadrillé (les petits carrés sont de coté 1). On part du coin supérieur gauche et on se déplace vers le coin inférieur droit. On ne peut, à chaque nœud, se déplacer que vers le bas ou la droite. On note A_n le nombre de chemins possibles. Par exemple, $A_1 = 2$ et $A_2 = 6$. On souhaite écrire une fonction qui calcule A_n pour tout entier n. Par exemple, pour $n = 2$:



Pour cela, on considère un problème plus général : on remplace le carré par un rectangle quadrillé de taille $n \times p$ où $(n, p) \in \mathbb{N}^2$ et on cherche à dénombrer le nombre de chemins du coin supérieur gauche vers le coin inférieur droit. On note $B_{n,p}$ le nombre de chemins possibles.

1. Calculer $B_{n,0}$ et $B_{0,p}$ pour tous $n, p \in \mathbb{N}^*$.
2. Montrer que, pour tout $(n, p) \in \mathbb{N}^2$, $B_{n,p} = B_{n-1,p} + B_{n,p-1}$.
3. Écrire une fonction `Calcul_B` récursive en pseudo-code prenant en entrée deux entiers `n` et `p` et calculant $B_{n,p}$.
4. Montrer que la fonction ainsi définie termine pour toute entrée dans $\mathbb{N}^2$.
5. En déduire une fonction `Calcul_A` prenant en entrée un entier `n` et calculant A_n.
6. Pour tout $(n, p) \in \mathbb{N}^2$, on note $C_{n,p}$ le nombre d'opérations arithmétiques nécessaires au calcul de $B_{n,p}$. Calculer $C_{n,0}$, $C_{0,p}$, puis déterminer une relation de récurrence vérifiée par $(C_{n,p})_{(n,p)\in\mathbb{N}^2}$.

7. En déduire une relation entre $C_{n,p}$ et$B_{n,p}$, pour tout $(n,p) \in \mathbb{N}^2$.
8. On fixe $(n,p) \in \mathbb{N}^2$. Dans le rectangle de format $n \times p$, nous représentons un chemin quelconque par une suite de mouvements vers le bas (B) ou vers la droite (D). Quel que soit le chemin, il y a n B et p D dans la suite correspondante. Réciproquement, une suite de ce type peut être associée à un chemin. Déduire de ce qui précède la valeur de $B_{n,p}$, puis de A_n.
9. En déduire une estimation de la complexité de la fonction calculant A.
   Indication : *on pourra utiliser l'équivalent de Stirling :* $n! \underset{n\to+\infty}{\simeq} \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$.
10. À l'aide de la formule trouvée, écrire une fonction non récursive calculant A_n, pour tout entier n, de complexité $\underset{n\to+\infty}{\mathcal{O}}(n)$.