Nom :

Prénom :

Groupe:

Université de Nice-Sophia Antipolis
École Supérieure en Sciences Informatiques

2004–2005

Contrôle de Mathématiques Discrètes
du 24 Janvier 2005

Durée : 2 heures

1	
2	
3	
4	
5	

Tous documents autorisés. Toutes vos réponses doivent être prouvées. Si vous pensez que le texte d'une question est ambigu, faites une hypothèse raisonable et écrivez la sur votre copie. Le barême, donné à titre indicatif, tiendra compte de la qualité de la rédaction.

1 Définition inductive d'un ensemble de mots [4 points]

On considère l'ensemble M des mots sur l'alphabet $\{a,b,c\}$ tels que toutes les suites (maximales) de c consécutifs soient de longueur paire. Ainsi, le mot *abccccbaccaa* est un mot de M , mais le mot *aaccbbcccba* n'est pas un mot de M. On considère l'ensemble E défini inductivement par :

- $\epsilon \in E$
- Si $m \in E$, alors $am \in E$
- Si $m \in E$, alors $bm \in E$
- Si $m \in E$, alors $ccm \in E$

1. A-t-on M inclus dans E?

2. A-t-on E inclus dans M?

3. Le schéma définissant E est-il libre?

4. Trouver et résoudre une relation de récurrence permettant de compter le nombre de mots de longueur n appartenant à E.

2 Tours de Hanoï, avec trois piquets dont un piquet central [4 points]

On considère une variante des tours de Hanoï avec trois piquets A,B,C et n disques.

- les n disques sont de tailles différentes
- initialement les n disques sont rangés sur un des piquets X, du plus grand en bas, au plus petit en haut
- on désire amener les n disques dans le même ordre sur un autre piquet Y
- on ne peut déplacer qu'un seul disque à la fois
- on ne peut pas poser un disque sur un disque de taille inférieure
- tous les déplacements de disque doivent se faire via le piquet central (un déplacement direct entre les deux piquets extérieurs est interdit)

On rappelle que si X et Y sont les piquets extérieurs, le nombre minimum de déplacements pour transporter n disques entre X et Y est de $3^n - 1$.

On suppose que moveExterieurToCentral(n,X) (que vous pourrez noter mE2C(n,X) pour abréger) déplace de manière optimale une tour de hauteur n d'un piquet extérieur X vers le piquet central. De même, moveCentralToExterieur(n,X) (que vous pourrez noter mC2E(n,X) pour abréger) déplace de manière optimale une tour de hauteur n du piquet central vers un piquet extérieur X.

1. Déterminer, prouver et résoudre les équations de récurrence permettant de calculer la complexité de ces deux méthodes.

2. En déduire que l'algorithme suivant est optimal

```
moveExterieurToExterieur(n, initial, final){
  if (n>0){
        mE2C(n-1, initial);
        mC2E(n-1, final);
        moveDisque(initial, central);
        mE2C(n-1, final);
        mC2E(n-1, initial);
        moveDisque(central, final);
        mE2C(n-1, initial);
        mC2E(n-1, final);
        }
  }
```

3 Tours de Hanoï, avec quatre piquets [4 points]

On suppose connue la méthode moveHanoi(n,initial, final, intermediaire) qui déplace de manière optimale (c'est à dire avec $2^n - 1$ mouvements de disques) une tour de Hanoi de hauteur n, d'un piquet initial vers un piquet final en utilisant un seul piquet intermédiare avec comme contraintes :

- les n disques sont de tailles différentes
- initialement les n disques sont rangés sur un piquet initial, du plus grand en bas, au plus petit en haut
- on désire amener les n disques dans le même ordre sur un piquet final

- on ne peut déplacer qu'un seul disque à la fois
- on ne peut pas poser un disque sur un disque de taille inférieure.

On suppose que l'on dispose d'un quatrième piquet. Pour chacune des stratégies suivantes, évaluer la compléxité en nombre de déplacements de disques, et déterminer la meilleure de ces stratégies.

- Stratégie 1 : Déplacer les n-2 plus petits disques vers l'un des piquets intermediaires, puis déplacer les 2 grands disques restant du piquet initial vers le piquet final (en utilisant le piquet intermediaire resté libre) , enfin déplacer les n-2 plus petits disques disques vers le piquet final.

- Stratégie 2 : Déplacer les n-3 plus petits disques vers l'un des piquets intermediaires, puis déplacer les 3 grands disques vers le piquet final (en utilisant le piquet intermediaire resté libre) , enfin déplacer les n-3 disques vers le piquet final.

- Stratégie 3 : Déplacer les n/2 plus petits disques vers l'un des piquets intermediaires, puis déplacer les n/2 grands disques vers le piquet final (en utilisant le piquet intermédiaire resté libre), enfin déplacer les n/2 petits disques vers le piquet final.

- Meilleure stratégie parmi les trois?

Question facultative : avez-vous une meilleure stratégie ?

4 Listes d'entiers [4 points]

On considère des listes d'entiers L_E, définies inductivement par :

- $liste\_vide \in L_E$
- Si $l \in L_E$ et $e \in E$, alors $cons(e,l) \in L_E$

On rappelle que les fonctions $car(l)$ et $cdr(l)$ sont définies sur les listes non vides par :

Si $l = cons(e,l')$, $car(l) = e$ et $cdr(l) = l'$

On dit qu'une liste l est croissante, si elle est vide ou réduite à un seul élément ou sinon si $car(l)$ est inférieur ou égal à tous les éléments de $cdr(l)$, et $cdr(l)$ est elle même une liste croissante. Si de plus il n'y a pas deux éléments consécutifs égaux de la liste l, l est strictement croissante.

Soit LC_E le sous ensemble de L_E constitué des listes croissantes, et LSC_E le sous-ensemble de LC_E constitué des listes strictement croissantes.

Donner une définition *récursive* des fonctions suivantes et déterminer la complexité de chacune de ces fonctions, en supposant que car, cdr et $cons\ estVide$ sont toutes en temps constant.

1. Croissante : $L_E \rightarrow \{vrai,faux\}$
   Cette fonction retourne vrai si et seulement si la liste considérée est croissante

2. Simplifiée : $LC_E \to LSC_E$
   Cette fonction retourne une liste où les répétitions ont été éliminées (donc une liste strictement croissante)

3. Fusion : $LC_E \times LC_E \to LC_E$
   Cette fonction retourne une liste de LC_E qui est la fusion des deux listes triées.

4. Intersection : $LSC_E \times LSC_E \to LSC_E$
   Cette fonction retourne une liste de LSC_E contenant les éléments qui sont dans les deux listes.

5 Équations de récurrence [4 points]

Donnez l'ordre de grandeur des solutions des équations de récurrence suivantes :

1. $u_n = 6u_{n-1} - 8u_{n-2} + 3^n$

2. $v_n = 6v_{n-1} - 8v_{n-2} + 4^n$

3. $w_n = 6w_{n-1} - 8w_{n-2} + 2^n$

4. $x_n = 6x_{n-1} - 9x_{n-2} + 3^n$