L3 Informatique **année 2010-2011**

Méthodes Algorithmiques

Examen du 19 Avril 2011
Responsable : Sophie Pinchinat

Les documents sont autorisés. Le barème est donné à titre indicatif.

Diviser-pour-Régner (7 points)

Soit $A[1 \dots n]$ un tableau de n éléments d'un type t muni d'une relation d'ordre totale entre les éléments (par exemple le type des réels, des caractères avec l'ordre du code ASCII, etc.).

1. Expliquez, en donnant un algorithme, comment il est possible de déterminer l'indice d'un plus grand élément de A en $n-1$ comparaisons (d'éléments); et, subséquemment, comment déterminez l'indice d'un plus petit élément en $n-2$ comparaisons.

   Nous avons ainsi obtenu un algorithme qui permet de calculer le plus petit et le plus grand élément d'un tableau de taille n en $2n-3$ comparaisons.

2. Proposez un algorithme basé sur la technique *Diviser-pour-Régner* capable de déterminer les indices d'un plus petit et d'un plus élément d'un tableau de n éléments en faisant moins de $2n-3$ comparaisons.

   Quel est le nombre exact de comparaisons que votre algorithme requiert (On pourra supposer que n est une puissance de 2.)? Justifiez votre réponse.

Tri (4 points)

On considère la procédure de tri fusion étudiée dans le cours.

1. Dessiner l'arborescence récursive de la procédure sur un tableau de 16 éléments.
2. Expliquez pourquoi une solution basée sur la méthode de programmation dynamique ne permet pas d'améliorer la vitesse d'un bon algorithme diviser-pour-régner tel que celui du tri fusion.

Une variante du problème des reines (9 points)

On considère un échiquier de taille $n \times n$, avec $n > 1$, dont on désigne les cases par des couples (i, j) où i est le numéro de ligne et j le numéro de la colonne. On s'intéresse à trouver le nombre minimal de reines à placer (et comment les placer) de telle façon que chaque case libre soit attaquée par au moins une reine.

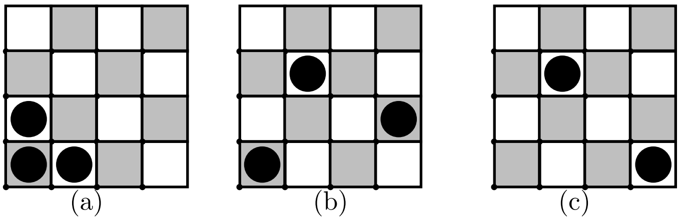

Figure 1: Des solutions optimales pour $n = 4$

Pour $n = 3$, une reine suffit en la plaçant dans la case $(2, 2)$ de l'échiquier, et cette solution est évidemment optimale. Pour $n = 4$, il y a plusieurs solutions (à 3 reines, mais aussi à 2 reines) dont certaines sont représentées en Figure 1.

Ainsi, et contrairement au problème classique des n reines, les reines peuvent se prendre mutuellement, et ne sont donc pas *a priori* sur des lignes différentes, comme le montre le cas de la Figure 1(a). On conviendra qu'une solution partielle est une suite de cases $\{(i_1, j_1), (i_2, j_2), \ldots (i_p, j_p)\}$, où on a déjà placé des reines.

1. Après avoir déterminé une borne pour la longueur de la liste correspondant à une solution totale, proposez un algorithme glouton qui calcule une solution qui n'est pas nécessairement optimale.

2. Proposez maintenant une méthode basée sur les *Essais Successifs* pour parcourir toutes les solutions possibles (non nécessairement optimales). On convient d'appeler `ResteCaseLibre` le prédicat sur les solutions qui indique qu'une solution n'est pas totale, c-à-d. qu'il existe une case de l'échiquier qui n'est attaquée par aucune reine.

3. Adaptez la méthode développée au Point 2 pour calculer une solution optimale. Si vous utilisez une technique d'élagage, décrivez-la soigneusement.

4. *(Question bonus)* On attribue à chaque case de l'échiquier une pondération (c-à-d. un nombre qui indique combien de cases sont attaquées par une reine placée dans cette case). Utilisez ces données pour proposer une heuristique et pour déterminer les valeurs d'une fonction d'évaluation. Utilisez cette fonction pour proposer un algorithme de type *séparation et évaluation progressive (Branch and Bound)* qui calcule une solution optimale. Vous prendrez soin de définir précisément l'heuristique que vous utilisez, et de la justifier.