EXAMEN ALGORITHMIQUE ET COMPLEXITE
Master 1 Informatique
Janvier 2011.

Rédigez sur 2 copies séparées.

PARTIE A. Sur 10 points. Corrigée par JJ Chabrier. 2 points par question.

A1. Donner la définition d'un CSP (Problème de Satisfaction de Contraintes).

A2. Définir le problème des N-reines sous forme de CSP.

A3. Citer 2 méthodes de résolution de CSP.

A4. Quel est l'intérêt des matrices totalement unimodulaires pour la programmation linéaire ?

A5. Quelle est la complexité d'un algorithme optimal de tri, qui compare entre eux les n éléments à trier ?

PARTIE B. Sur 10 points. Corrigée par D. Michelucci.

B1 - Citez au moins 2 problèmes solubles par la programmation dynamique, par exemple ceux vus en TP (programmés en ocaml). Noté sur 1 point.

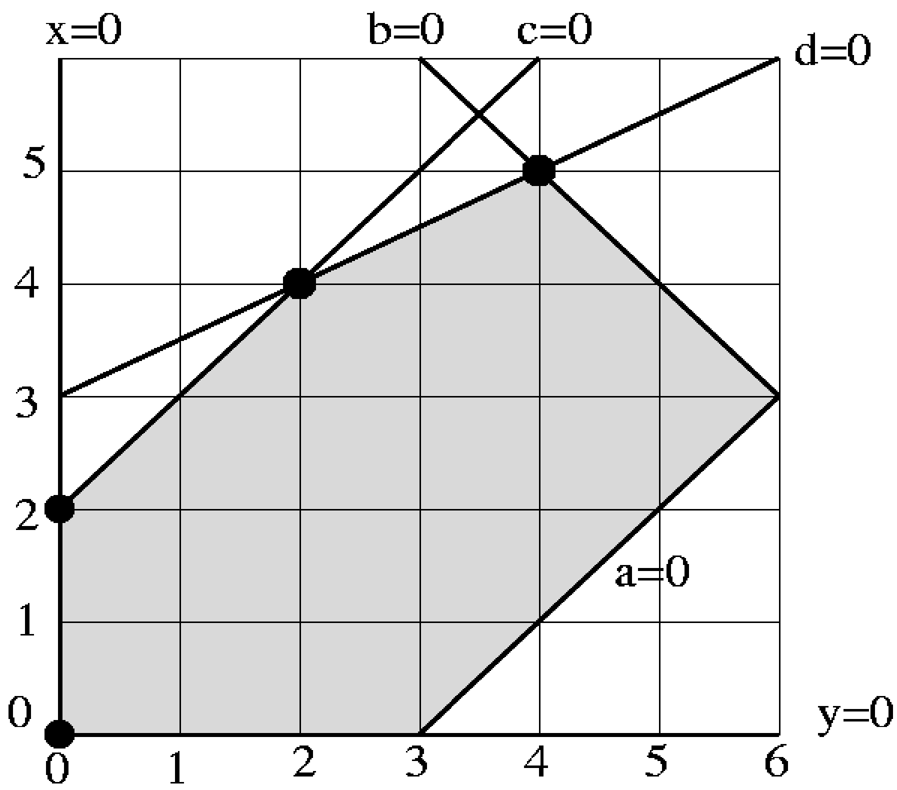
B2- Ecrire en ocaml la fonction: *interpol (x1,y1) (x2,y2) x* ; tous les paramètres sont des nombres flottants; la fonction doit retourner le nombre flottant y tel que le point (x, y) appartienne à la droite passant par les points (x1, y1) et (x2, y2). Vous supposerez que (x1,y1) est différent de (x2,y2). Noté sur 1 point.

B3- Résoudre par la méthode du simplexe le problème de programmation linéaire suivant, en écrivant toutes les étapes. Vous appellerez les variables d'écart a, b, c, d. Faites un dessin du polytope dans le plan Oxy. La matrice est-elle totalement unimodulaire ? Pourquoi ? On rappelle que toutes les variables (x, y, a, b, c, d) doivent être positives ou nulles, même s'il est d'usage de ne pas le préciser dans l'énoncé. Noté sur 8 points.

```
max y
x-y <= 3
x+y <= 9
-x+y <= 2
-x+2y <= 6
```

B.3 Résoudre le problème de programmation linéaire suivant avec a méthode du simplexe:

max y
x-y <= 3
x+y <= 9
-x+y <= 2
-x+2y <= 6



Après ajout des variables d'écart a, b, c, d, le premier tableau (sommet x=0,y=0 sur la figure) est:

max y *** augmenter y améliorera l'objectif
a= 3 – x + y
b= 9 – x - y
c= 2 + x - y *** si y dépasse 2, c devient négatif
d= 6 + x – 2y

En pivôtant sur la variable y et la ligne c, le sommet x=0,y=2 est atteint, dans le tableau :

max 2 + x – c *** augmenter x améliorera l'objectif
a= 5 - c
b= 7 -2 x + c
y= 2 + x - c
d= 2 – x + 2c *** si x dépasse 2, d devient négatif

En pivôtant sur la variable x et la ligne d, le sommet x=2,y=4 est atteint dans le tableau:

max 4 + c – d *** augmenter c améliorera l'objectif
a= 5 – c
b= 3 – 3 c + 2 d *** si c dépasse 1, b devient négatif
x= 2 + 2 c – d
y= 4 + c – d

En pivôtant sur la variable c et la ligne b, le sommet optimum x=4,y=5 est atteint:

max 5 – 1/3 b – 1/3 d *** l'optimum est atteint: les coefficients de b et d sont négatifs

c= 1 – 1/3 b + 2/3 d
a= 4 + b/3 – 2/3 d
x= 4 – 2/3 b + 1/3 d
y= 5 – 1/3 b – 1/3 d

Donc l'optimum:y = 5 est atteint en x=4, y=5, a=4, c=1, b=d=0.

La matrice n'est pas totalement unimodulaire, pour beaucoup de raisons :

- il y a un coefficient 2, dans le membre gauche de: x+2y <= 6 (donc différent de 0, 1, -1)
- les droites d'équations b=0 et c=0 se coupent en un point non entier (x=7/2, y=5/2)
- la pente de la droite d=0 vaut 1/2 ; seules les pentes 0, +1, -1 sont possibles dans le cas TU
- dans le tableau final, la matrice contient des coefficients + ou - 1/3, + ou - 2/3 différents de 0, 1, -1.

B.1 Ont été résolus par programmation dynamique en TP: le parenthésage optimal d'un produit de matrices, le sac à dos avec poids entiers, la plus longue sous séquence commune entre deux séquences, les dates au plus tôt et au plus tard dans un graphe de tâches.

B.2 Beaucoup de solutions sont possibles. En voici une.

```
ocaml
# let interpolate (x1,y1) (x2,y2) x = (* gives y such that (x,y) lies on the line *)
      y1 +.  (x -. x1) *. (y2 -. y1)  /. (x2 -. x1) ;;
  val interpolate : float * float -> float * float -> float -> float = <fun>
# interpolate (100., 30.) (300., 90.) 200. ;;
- : float = 60.
# interpolate (100., 30.) (300., 90.) 100. ;;
- : float = 30.
# interpolate (100., 30.) (300., 90.) 300. ;;
- : float = 90.
```

A.4 L'intérêt des matrices totalement unimodulaires en PL:

La solution X au problème de programmation linéaire :

max C . X avec A X <= B

est entière quand la matrice A est totalement unimodulaire et que le vecteur B est entier (et que le polytope réalisable est non vide, évidemment...). La méthode du simplexe fournira cette solution optimale entière – ou une des solutions optimales entières s'il y a un continuum de solutions.

Si le vecteur B est entier, et que A est entière mais non totalement unimodulaire, alors l'optimal n'est pas forcément entier, et trouver le meilleur point entier est un problème beaucoup plus difficile, en haute dimension. Exemple: dans le problème B.3, supprimer la contrainte d >= 0; alors le sommet avec le y maximal devient (x=3/2, y=11/2), qui n'est pas entier : c'est le point d'intersection des droites d'équation b=c=0 (voir Figure); le point entier ayant le y maximum est x=3, y=5 ou x=4,y=5. Ici il suffit de tronquer le sommet réel optimal pour trouver le sommet entier optimal, mais ce n'est pas vrai en général.