EXAMEN D'ALGORITHMIQUE, M1, Janvier 2014

Une page A4 recto-verso autorisée. Calculatrice, téléphone, ordinateur, lunettes Google interdits. N'écrivez aucun programme. Ecrivez lisiblement. Répondez aux questions dans l'ordre, en indiquant le numéro de chaque question.

1. Citez un problème indécidable en informatique.

2. Citez en un deuxième.

3. Citez un problème informatique pour lequel aucun algorithme en temps polynomial n'est connu.

4. Citez en un deuxième.

5. Citez un algorithme optimal de tri, qui procède par comparaisons ; quel est l'ordre de grandeur du nombre de comparaisons effectuées pour trier n éléments ?

6. Citez en un autre.

7. En TP, nous avons affiché des courbes définies par leurs équations. Comment avons nous représenté ces équations ?

8. (suite) Comment avons nous fait pour décider que certains rectangles ne pouvaient pas être traversés par la courbe ?

9. En TP, nous avons calculé les racines comprises entre 0 et 1 de polynômes univariés. Quel est le principe de la méthode que nous avons utilisée ?

10. Vous disposez d'une arithmétique sur de grands entiers relatifs, qui fournit l'addition, la soustraction, le produit, le quotient, le modulo, la comparaison, la conversion à partir d'unechaîne de caractères, ou vers une chaîne de caractères ; mais elle ne fournit pas la partie entière de la racine carrée. Proposez un algorithme raisonnable (en temps polynomial avec le nombre de chiffres de a) pour calculer le grand entier $x \geq 0$ tel que $x^2 \leq a < (x+1)^2$, avec a un grand entier donné. Ceci exclut l'algorithme qui essaie tous les entiers 0, 1, 2, ... jusqu'à trouver la solution.

11. (suite) Proposez un deuxième algorithme, de principe différent.

12. En TP, nous avons utilisé la programmation dynamique pour résoudre deux problèmes. Citez un de ces problèmes.

13. (suite) Citez l'autre.

14. La méthode de Knuth-Bendix a terminé avec succès sur ces trois règles de ré-écriture : $AA \to \epsilon$; $BB \to \epsilon$; $BAB \to ABA$. On rappelle que ϵ est le mot vide. Listez les mots (formés des lettres A et B) irréductibles. Disposez les sous forme arborescente.

15. Vous disposez d'une balance à deux plateaux, de haute précision. Vous avez 3 pièces d'or ; l'une d'elles est fausse (exactemement une), et plus lourde.

Comment, en une seule pesée, pouvez vous déterminer quelle est la pièce fausse ? Rédigez votre méthode avec soin.

16. (suite). Vous avez 3^k pièces d'or ; exactement une des 3^k pièces d'or est fausse et plus lourde. Comment, en k pesées, pouvez vous déterminer quelle est la pièce fausse ? Rédigez votre méthode avec soin.

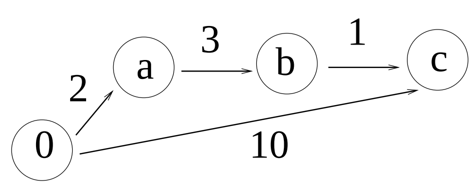
17. Déroulez les trois phases du tri par base ("radix sort") sur cet ensemble d'entiers : 888, 414, 148, 814, 881, 114, 118, 811, 844, 488, 814. N'utilisez que les trois tiroirs évidents.

18. Déroulez l'algorithme d'Euclide étendu (ou Bézout) pour $a = 100$ et $b = 46$ en remplissant un tableau comme ci-dessous. L'algorithme vu en cours calcule g le PGCD de a et b, ainsi que deux entiers relatifs ($\mathbb{Z}$) u et v. u et v sont tels que $au + bv = g = PGCD(a, b)$; on note $r = a \mod b$, et $q = \lfloor \frac{a}{b} \rfloor$ le quotient de a par b. Il n'y a qu'une seule réponse correcte.

a	b	r	q	g	u	v
100	46					
?						
?						
?						
2	0	indéf	indéf	2	1	0

19. (suite) Quand vous remplissez les cases des colonnes u et v, en fonction du contenu de la ligne d'après ou d'avant, quelles formules utilisez vous ? Réponse suggérée : soient (g', u', v') le contenu de la ligne en dessous (ou au dessus ?). Alors $(g, u, v) = (g', ?, ?)$. La preuve n'est pas demandée. Il n'y a qu'une seule réponse correcte.

20. Posez le problème du plus court chemin entre le sommet étiqueté 0 et le sommet étiqueté c comme un problème de programmation linéaire, avec la méthode du potentiel (a, b, c sont les potentiels). Utilisez d'abord des inéquations, puis utilisez des variables d'écart.



21. (suite) Résolvez ce problème par la méthode du simplexe.

22. Quel problème résout la méthode appelée "union find" ?

23. Citez un problème ou un algorithme vu en cours qui utilise l'"union find".

24. Certains problèmes, bien que décidables, sont si difficiles que l'on ne sait même pas vérifier leur solution en temps polynomial. Citez en un.