Algorithmes et Arbres, $2^{ème}$ Semestre 11/12

ALGORITHMES ET ARBRES

Durée : 2h, polycopiés et calculatrices autorisés.
Barême indicatif sur 20 : exo1(5 points), exo2(7 points), exo3(8 points).

1. Mise en bouche.

Exercice 1. Notations asymptotiques. Montrer que $\sum_{k=0}^{n} 2^k = \Theta(2^n)$.
Réponse : $\sum_{k=0}^{n} 2^k = 2^{n+1} - 1$ qui est compris entre 1×2^n et 2×2^n pour tout $n > 1$.
Exercice 2. Chercher les intrus. On dispose d'un ensemble de n pièces d'or toutes de même poids sauf deux défectueuses qui sont plus légères. Le problème consiste à retrouver les deux pièces défectueuses en utilisant une balance de type Roberval (balance à 2 plateaux qui permet de dire uniquement s'il y a un plateau plus lourd que l'autre).

(1) Quel est le nombre minimal de pesées dans le pire des cas pour trouver les 2 fausses pièces quand il y a $n = 3$ pièces ?
Réponse : Une suffit, on compare deux pièces $p1$ et $p2$, si $p1 < p2$ alors $p1$ et $p3$ sont les fausses pièces. Si $p1 > p2$ alors $p2$ et $p3$ sont les fausses pièces. Sinon si $p1 = p2$, alors $p1$ et $p2$ sont les fausses pièces.

(2) Même question quand il y a $n = 4$ pièces.
Réponse : Une pesée ne suffit pas car si on compare deux pièces $p1$ et $p2$ et qu'il y a égalité on ne peut conclure (soit $p1$ et $p2$ sont les fausses pièces, soit $p3$ et $p4$ sont les fausses pièces) et si on compare deux pièces $\{p1, p2\}$ avec $\{p3, p4\}$ et qu'il y a égalité, on ne peut conclure (on sait juste que $p1$ ou $p2$ est fausse et $p3$ ou $p4$ est fausse). Avec deux pesées, on y arrive, on compare $p1$ et $p2$
si $p1 = p2$, on compare $p1$ et $p3$: si $p1 < p3$ alors $p1$ et $p2$ sont les fausses pièces sinon $p3$ et $p4$ sont les fausses pièces
sinon on sait de $p1$ ou $p2$ qui est la fausse pièce et on compare $p3$ et $p4$ pour trouver l'autre fausse pièce.

On veut résoudre le problème avec un algorithme diviser pour régner. Pour cela, on divise l'ensemble de n pièces en trois sous-ensembles T_1, T_2 et T_3 contenant respectivement :
– k, k et k pièces si $n = 3k$,
– k, k et $k + 1$ pièces si $n = 3k + 1$
– k, k et $k - 1$ pièces si $n = 3k - 1$

(3) Où sont les pièces défectueuses, si le premier tas est aussi lourd que le deuxième tas ? Il y a deux configurations possibles, quelle pesée faut-il faire pour lever l'ambiguité (distinguer les cas suivant n mod 3) ?
Réponse : Soit il y a une fausse pièce dans $T1$ et une fausse pièce dans $T2$, soit les 2 fausses pièces sont dans $T3$. Pour lever l'ambiguité, on compare $T1$ avec $T3$ en rajoutant, s'il le faut, à $T1$ ou $T3$ une pièce de $T2$ pour équilibrer le nombre de pièces. Si cela penche vers $T1$, alors il y a 2 fausses pièces dans $T3$. Sinon il y a une fausse pièce dans $T1$ et une dans $T2$ (même s'il y a égalité, on a alors rajouté la fausse pièce de $T2$ à $T3$!).

(4) Où sont les pièces défectueuses, si le premier tas est moins lourd que le deuxième tas ? Il y a deux configurations possibles, quelle pesée faut-il faire pour lever l'ambiguité (distinguer les cas suivant n mod 3) ?
Réponse : Soit il y a une fausse pièce dans $T1$ et une dans $T3$, soit il y 2 fausses pièces dans $T1$. Pour lever l'ambiguité, on compare $T1$ avec $T3$ en rajoutant, s'il le faut, à $T1$ ou $T3$ une pièce de $T2$ pour équilibrer le nombre de pièces. Si cela penche vers $T3$ il y a deux fausses pièces dans $T1$ sinon il y a une fausse pièce dans $T1$ et une dans $T3$ (il y a équilibre).

(5) En déduire une induction pour le nombre de pesées $C_2(3^n)$ effectuées dans le pire des cas pour trouver les 2 fausses pièces dans un ensemble de 3^n pièces (On notera $C_1(n)$ le nombre de pesées pour trouver une fausse pièce dans un ensemble de taille n où il y a une fausse

pièce et $C_2(n)$ le nombre de pesées pour trouver deux fausses pièces dans un ensemble de taille n où il y a deux fausses pièces.
Réponse : $C_2(3^n) = \max(C_2(3^{n-1}), 2C_1(3^{n-1})) + 2$. En fait, on fait 2 pesées et après on recommence avec soit un ensemble de $n/3$ pièces qui contient 2 fausses pièces, soit il y 2 ensembles de taille $n/3$ qui contiennent chacun une fausse pièce.

(6) On rappelle que $C_1(3^n) = n$. Montrer par induction que $C_2(3^n) = 2n$. Que peut-on conjecturer sur le comportement asymptotique de $C_2(n)$?
Réponse : $C_2(3) = 1$ (vu avant) et par induction $C_2(3^n) = \max(C_2(3^{n-1}), 2C_1(3^{n-1})) + 2 = \max(2(n-1), 2(n-1)) + 2 = 2n$. Il semble être en $\Theta(\log(n))$.

Exercice 3. Analyse de complexité. Voici le pseudo-code permettant de calculer le n-ième terme de la suite récurrente $f(n) = nf(n-1) + f(n-2)$ et $f(0) = f(1) = 1$.

Algorithme 1: CalculF

Entrées : un entier n
Sorties : la valeur de $f(n)$
si $n < 2$ **alors**
  Renvoyer 1
sinon
  Renvoyer $n\times$CalculF$(n-1)$+CalculF$(n-2)$

(1) Calculer $f(3)$.
Réponse : 10.

(2) On note $C(n)$ le nombre d'additions faites pas CalculF pour renvoyer $f(n)$. Donner une récurrence pour $C(n)$.
Réponse : $C(n) = 1 + C(n-1) + C(n-2)$

(3) Montrer que pour tout $n > 5, C(n) > (\frac{3}{2})^n$. En déduire un Ω pour la complexité en nombre d'additions de CalculF.
Réponse : $C(6) = 12 > (\frac{3}{2})^6 \simeq 11.39$ et $C(7) = 20 > (\frac{3}{2})^7 \simeq 11.39$, puis par induction si $C(n-1) > (\frac{3}{2})^{n-1}$ et $C(n-2) > (\frac{3}{2})^{n-2}$ alors $C(n) > 1 + \frac{5}{2}(\frac{3}{2})^{n-2} > \frac{9}{4}(\frac{3}{2})^{n-2}$.
$C(n) = \Omega((\frac{3}{2})^n)$

Exercice 4. le meilleur et le second. Soit $T[1..n]$ un tableau contenant vos notes de partiel. On cherche un algorithme efficace permettant de trouver la meilleur note et la deuxième meilleur note (qui peut être égale à la première).

(1) Si on procède en cherchant d'abord le max $M1$, puis le max $M2$ dans le tableau privé du max $M1$, combien de comparaisons a-t-on fait ?
Réponse : $2n-3$

(2) On divise le tableau en 2 tableaux, $T1 := T[1..\lfloor n/2 \rfloor]$ et $T2 := [\lfloor n/2 \rfloor + 1, .., n]$, notons M_a le max de $T1$ et M_b le max de $T2$. Indiquer suivant que M_a est plus petit ou plus grand que M_b qu'elle opération reste à faire pour trouver $M1$ et $M2$.
Réponse : Si $M_b < M_a$, $M_a = M1$ et le deuxième max est le plus grand entre M_b et le deuxième max du tableau $T1$. Sinon $M_b = M1$ et le deuxième max $M2$ est le plus grand entre M_a et le deuxième max du tableau $T2$.

(3) Combien de comparaisons fait cet algorithme pour un tableau de taille $2n$?
Réponse : $n-1$ pour trouver M_a, $n-1$ pour trouver M_b, une comparaison entre M_a et M_b, et $n-2$ pour trouver le deuxième max $M3$ de $T1$ ou $T2$ et une dernière comparaison pour comparer $M3$ avec M_a ou M_b. Soit au total $3n-2$ comparaisons.