TD 06 – Partiel de l'an dernier

Tous les exercices sont indépendants. Chaque algorithme donné doit être accompagné d'une explication (assez brève) pour montrer qu'il renvoie la bonne réponse, et qu'il a bien le coût demandé dans l'énoncé. Mais ne perdez pas de temps à écrire des pseudo-programmes détaillés, c'est inutile. That said, enjoy and good luck!

Exercice 1. *Maximum d'un tableau trié mais circulairement décalé*

On considère un tableau A de n entiers distincts triés qui a subi un décalage circulaire de k positions sur la droite. Par exemple $A = [35, 42, 5, 15, 21, 26]$ a été décalé de deux positions. On donne A mais on ne connaît pas la valeur de k.

1. Donner un algorithme de coût $O(\log n)$ pour déterminer la position du maximum de A.

Exercice 2. *Eléments dominants*

Soit S un ensemble de n points du plan (x_i, y_i), $1 \leq i \leq n$, à coordonnées entières distinctes deux à deux. On dit qu'un point (x, y) domine un point (x', y') si $x > x'$ et $y > y'$. Un point de S est dit maximal s'il n'est dominé par aucun point de S.

1. Proposer un algorithme efficace pour déterminer les points maximaux de S.

Exercice 3. *Retour au gymnase (take 1)*

On reconsidère le problème de planifier n évènements sportifs. L'évènement E_i doit avoir lieu dans l'intervalle $[d_i, f_i[$. Mais au lieu de chercher à caser un nombre maximal d'évènements dans un seul gymnase (comme on a fait en cours), on cherche à les caser tous, cette fois en minimisant le nombre de gymnases nécessaires. Deux évènements dont les intervalles ne sont pas disjoints doivent être planifiés dans deux gymnases différents.

1. Proposer un algorithme pour résoudre ce problème, et justifier sa correction.

Exercice 4. *Partition en tranches*

On considère un n-uplet $S = (s_1, s_2, \ldots, s_n)$ d'entiers, qu'on veut découper, sans le ré-ordonner, en k tranches. L'objectif est de minimiser m_k, la plus grande des sommes partielles des k tranches. Par exemple si $S = (2, 5, 3, 8, 6, 1)$ et $k = 3$, on prendra la solution $(2, 5, 3)|(8)|(6, 1)$, avec $m_3 = 10$.

1. Proposer un algorithme efficace pour calculer le meilleur m_k.

Exercice 5. *Retour au gymnase (take 2)*

On revient au problème du gymnase unique et des n évènements $E_i = [e_i, f_i[$, où e_i est l'heure de début et f_i celle de fin. On a vu en cours un algorithme glouton pour trouver le nombre maximal d'évènements sans intersection dans le temps. Maintenant, chaque évènement E_i a une valeur v_i, et on cherche un algorithme pour trouver la somme maximale des valeurs d'évènements sans intersection dans le temps.

1. Donner un exemple où l'algorithme glouton du cours ne renvoie pas la valeur maximale.
2. Donner un algorithme de programmation dynamique pour résoudre le problème. Quelle est sa complexité?

Exercice 6. *Multiplication de polynômes*

L'objet de cet exercice est la multiplication rapide de deux polynômes. On appele *n-poly* un polynôme de degré $n - 1$, donc avec n coefficients. Soient $P = \sum_{i=0}^{n-1} a_i X^i$ et $Q = \sum_{i=0}^{n-1} b_i X^i$ deux n-polys. Leur produit $R = P \times Q$ est un $(2n - 1)$-poly. On note $M(n)$ (resp. $A(n)$) le nombre de multiplications (resp. d'additions) entre coefficients pour multiplier deux n-polys.

1. Montrer que $M(n) = n^2$ et $A(n) = (n-1)^2$ pour l'algorithme usuel

2. On suppose n pair, $n = 2 \times m$. On écrit $P = P_1 + X^m \times P_2$ et $Q = Q_1 + X^m \times Q_2$, où P_1, P_2, Q_1, et Q_2 sont des m-polys. On définit $R_1 = P_1 \times Q_1$, $R_2 = P_2 \times Q_2$, et $R_3 = (P_1 + P_2) \times (Q_1 + Q_2)$ (donc R_1, R_2, et R_3 sont des $(n-1)$-polys) :

   (i) Exprimer $R = P \times Q$ en fonction de R_1, R_2, et R_3

   (ii) Calculer $M(n)$ et $A(n)$, en supposant utiliser l'algorithme usuel pour calculer R_1, R_2, et R_3.

3. On suppose maintenant que $n = 2^s$ et on utilise récursivement l'approche précédente. Proposer une formule de récurrence pour $M(n)$ et $A(n)$ avec cet algorithme (dû à Monsieur Karatsuba).

4. Calculer la valeur du $M(n)$ précédent.

5. **(difficile)** Calculer la valeur du $A(n)$ précédent (nécessite d'avoir lu le poly sur le calcul des récurrences)