

Examen 3h

15 Janvier 2025

- Exercice 1 - Questions de cours.

1. Expliquer pourquoi on peut construire un tas en temps linéaire.
2. Proposer un algorithme de résolution de 3-SAT en temps $O^*(c^n)$ avec $c < 2$ et n est le nombre de variables.

- Exercice 2 - Question Bonus. Dans cet examen, vous trouverez quatre questions (Paradigme i) pour $i = 1, 2, 3, 4$. Dans cet exercice réalisez la bijection entre les Paradigmes i et les quatre items "NP-difficile", "Polynomial avec un algorithme diviser pour régner", "Polynomial avec un algorithme glouton", "Polynomial avec de la programmation dynamique". Aucune justification n'est demandée ici. *Cet exercice est noté ainsi : 2 points si la permutation est exacte, 1 point si l'un des paradigmes est bien associé, 0 sinon. L'espérance est de $2/3$ de points si vous répondez au hasard.*

- Exercice 3 - Chemin Hamiltonien. Un *tournoi* $T = (V, E)$ est un graphe orienté tel que pour toute paire de sommets distincts $x, y \in V$, l'une des deux arêtes xy ou yx est dans T . Un *chemin hamiltonien* est une énumération $v_1, \dots, v_n$ telle que $v_i v_{i+1}$ est une arête pour tout $i = 1, \dots, n-1$.

1. (Paradigme 1) Quelle est la complexité de calculer un chemin hamiltonien dans un tournoi T sur n sommets, ou de renvoyer FAUX si aucun n'existe ?
2. On se donne à présent une fonction de poids ω sur les arêtes de T . Le *poids* d'un chemin hamiltonien est la somme des $\omega(v_i v_{i+1})$, pour $i = 1, \dots, n-1$. Montrer que calculer un chemin hamiltonien de poids minimal dans un tournoi T est NP-difficile.

- Exercice 4 - Permutation.

1. Proposer un algorithme qui tire une permutation aléatoire uniforme de $\{1, \dots, n\}$.
2. Proposer un algorithme qui associe à toute permutation σ de $\{1, \dots, n\}$ un entier p_σ entre 0 et $n! - 1$ (codage), ainsi que l'algorithme de décodage prenant en entrée un tel entier p_σ et retournant la permutation σ .

- Exercice 5 - Sous-tableau. On se donne en entrée un tableau $T[1, \dots, n]$ ainsi qu'un entier k .

1. Un sous tableau T' de T réduit aux indices $i_1 < i_2, \dots < i_t$ est *k-propre* si $i_{j+1} - i_j \geq k$ pour tout $j = 1, \dots, t-1$. Proposer un algorithme de la meilleure complexité possible pour calculer un sous tableau *k-propre* dont la somme des entrées $T[i_j]$ est maximale.
2. (Paradigme 2) On se donne à présent une matrice $n \times n$ de pénalités $p(i, j)$ et l'on cherche à calculer un sous tableau dont les entrées sont $i_1 < i_2, \dots < i_t$ qui maximise la somme de ses entrées moins la somme des pénalités $p(i_j, i_{j+1})$ pour $j = 1, \dots, t-1$. Quelle est la complexité de ce problème ?

3. (Paradigme 3) Sous les mêmes hypothèses, on cherche à calculer un sous tableau dont les entrées sont $i_1 < i_2, \dots < i_t$ qui maximise la somme de ses entrées moins la somme des pénalités $p(i_j, i_\ell)$ pour tous les $1 \leq j < \ell \leq t$. Quelle est la complexité de ce problème ?

- Exercice 6 - Ordonnancement. On se donne en entrée un ensemble de n tâches de temps de complétion respectifs $T = [t_1, \dots, t_n]$. On dispose de plus de m machines et le but est de répartir ces tâches sur ces machines afin de minimiser le temps total d'exécution. En d'autres termes, on veut partitionner les entrées de T en m sous-ensembles $T_1, \dots, T_m$ de sommes respectives $S_1, \dots, S_m$ en cherchant à minimiser le maximum des S_i . On note ce problème $\text{ORDO}(T, m)$.

1. Montrer que $\text{ORDO}(T, m)$ est NP-difficile pour $m = 2$. On pourra réduire depuis SOMME.
2. (Paradigme 4) Quelle est la complexité d'approximer $\text{ORDO}(T, m)$ à un facteur constant 2 ?

- Exercice 7 - Isolation. On se donne un ensemble non vide S de parties de $[n]$. On considère un tableau $T = [p_1, \dots, p_n]$ de valeurs entières entre 1 et $2n$ tirées aléatoirement uniformément et indépendamment. Chaque $X \in S$ a donc un poids p_X correspondant à la somme des p_i , pour $i \in X$. On note $\min$ la valeur minimum de p_X , pour $X \in S$. Le lemme d'isolation affirme qu'avec probabilité au moins $1/2$, il existe un unique élément $X \in S$ de poids $\min$.

Dans ce problème, nous montrerons d'abord ce lemme, puis nous verrons une application.

- a. Décomposons le tirage de T en tirant les $n - 1$ premières valeurs et en retardant le tirage de p_n . Notons m_1 le poids minimal d'une partie $X \in S$ qui ne contient pas n . Notons aussi m_2 le poids minimal d'une partie $X \in S$ qui contient n (en considérant que $p_n = 0$ car la valeur de p_n n'a pas encore été tirée). Montrer que la probabilité que $p_n = m_1 - m_2$ est au plus $1/2n$.
- b. Conclure la preuve du lemme d'isolation.
- c. On se donne $M = (m_{i,j})$ une matrice carrée $n \times n$ de 0 et 1 qui possède m entrées 1. On suppose qu'il existe une permutation σ telle que $m_{i,\sigma(i)} = 1$ pour tout $i = 1, \dots, n$. Montrer que si on remplace chaque entrée $m_{i,j} = 1$ de M par une valeur $2^{p_{i,j}}$ avec $p_{i,j}$ entier tiré uniformément et indépendamment entre 1 et $2m$, alors le déterminant de la matrice M' obtenue est non zéro avec probabilité au moins $1/2$.
- d. En déduire un algorithme randomisé qui décide si un graphe biparti possède un couplage parfait.
- e. On veut maintenant construire un tel couplage et non plus seulement décider son existence. On suppose que l'on a m machines parallèles (autant que les arêtes du graphe biparti). Proposer un algorithme randomisé parallèle de construction de couplage parfait tel que chaque machine calcule un seul déterminant.