IFT 2125 A12 **Geña Hahn**

Introduction à l'algorithmique

EXAMEN FINAL

le 7 janvier 2013

Durée : 165 minutes

Valeur: 45% de la note, 50% de la note prédoc

Directives:

- Aucune documentation n'est permise.
- Répondez sur le questionnaire dans l'espace libre qui suit chaque question. **L'espace alloué n'est aucune indication de la longueur de la réponse! Il est souvent beaucoup trop grand.**
- Sauf indication contraire, aucun point ne sera accordé pour une réponse, correcte ou pas, si elle n'est pas accompagnée d'une justification.
- Notez la différence entre *justifier* (argument rapide et court, peut-être intuitif) et *prouver* ou *démontrer* (argument détaillé).
- Vous pouvez vous servir de résultats vus en cours, en TP ou dans des livres à condition de les énoncer précisément – sauf pour les démontrer eux-mêmes. Vous pouvez aussi vous servir des résulats donnés dans l'annexe.
- Pour répondre à une question, vous pouvez également vous servir de résultats énoncés dans d'autres questions dans l'examen, même si vous ne les avez pas démontrés.
- Rappel: $\mathbb{N}$ est l'ensemble des entiers non négatifs, $\mathbb{R}$ l'ensemble des réels, $\mathbb{R}^{\geq 0}$ l'ensemble des réels non négatifs et $\mathbb{R}^{>0}$ l'ensemble des réels positifs. Aussi, $\lg n = \log_2 n$ et $\ln n = \log_e n$ (où e est la base du logarithme naturel). Finalement, $\log n$ est le logarithme "générique" : la base peut être n'importe quel $b \in \mathbb{R}^{>1}$.
- Les définitions utilisées sont celles du cours et du livre Brassard/Bratley et ce sont celles-ci qui s'appliquent dans cet examen. En particulier, les fonctions sont toutes dans $\mathcal{F} = \{f \mid f : \mathbb{N} \longrightarrow \mathbb{R}^{\geq 0}\}$.

1. ________ /20	5. ________ /20	9. ________ /20
2. ________ /20	6. ________ /20	10. ________ /20
3. ________ /20	7. ________ /20	11. ________ /20
4. ________ /20	8. ________ /20	12. ________ /20

Total: ________ /240

Nom: ______________________________ **Code permanent:** ____________________

☐ **cocher si cet examen fait partie de votre examen prédoctoral.**

1. (**20 x 1 point**) Pour chacun des énoncés suivants indiquez s'il est vrai ou non en encerclant OUI ou NON. *AUCUNE JUSTIFICATION N'EST NECESSAIRE.*

(1) $0(n^2) = 0(2^{2\log_3 n})$ OUI NON

(2) Un algorithme dont la complexité est dans $O(n \lg n)$ est toujours préférable à un autre qui résoud le même problème en $O(n^3)$. OUI NON

(3) L'algorithme de Floyd peut être modifié pour qu'il trouve la fermeture transitive d'un graphe orienté. OUI NON

(4) Si u est un point d'articulation d'un graphe G alors $G - u$ a exactement deux composantes connexes. OUI NON

(5) Le principe d'optimalité dit que *si S est une solution optimale à un problème de taille n alors une solution optimale pour ce problème de taille $2n$ peut être obtenue à partir de S.* OUI NON

(6) L'algorithme de Prim ne marche que si les poids des arêtes sont positifs. OUI NON

(7) Si $f \in O(g)$ alors $g \in \Omega(f)$. OUI NON

(8) $\Theta(|\sin(n)| + 2) = \Theta(17)$ OUI NON

(9) Si $f \in \Theta(g)$ alors $3^f \in \Theta(3^g)$ OUI NON

(10) Puisque la médiane d'un fichier de n clé peut être trouvée en temps linéaire, Quick-Sort peut être modifié pour qu'il trie ce même fichier en $O(n \lg n)$ comparaisons dans le pire des cas. OUI NON

(11) Si $f(n) = \Theta(g(n))$, alors $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c,\ c \in \mathbb{R}$. OUI NON

(12) Soit $T(n) = 5T(\frac{n}{3}) + \theta(n)$, $\theta(n) \in \Theta(n^2)$. Alors $T(n) \in \Theta(n^2)$ OUI NON

(13) Un monceau (heap) de n clés peut être construit en temps dans $O(\lg n)$. OUI NON

(14) La complexité de la fouille en profondeur est dans $O(n)$ pour un graphe avec n sommets et m arêtes. OUI NON

(15) $P \subseteq NP$ OUI NON

(16) Multiplier deux matrices de dimensions $n \times n$ chacune demande au moins n^3 multiplications. OUI NON

(17) Tout algorithme de tri fait au moins $n \lg n$ comparaisons sur un fichier de n clés. OUI NON

(18) Quel que soit le problème, on peut toujours trouver un algorithme glouton pour le résoudre. OUI NON

(19) Trouver l'ordre optimal de multiplications chaînées de matrices demande un temps exponentiel si toutes les matrices sont de la même taille. OUI NON

(20) Pour minimiser le temps d'exécution de toutes les tâches dans une queue donnée, il faut accomplir les tâches dans l'ordre de temps d'execution non-décroissant. OUI NON

2. (**10 + 10 points**) Justifiez DEUX de vos réponses à la question 1. **Votre réponse ne vaut rien si elle ne comprend pas l'énoncé de ce que vous prouvez!**

   - Question ....

   - Question ....

3. (**20 points**) Nous avons prouvé que que si t est une fonction éventuellement non-décroissante et f une fonction $b-$lisse alors $t \in O(f)$ quand $t(n) \in O(f(n)|n \in P_b)$ (pour un $b \in \mathbb{N}^{\geq 2}$). Afin d'avoir le même résultat pour $\Theta(f)$, il faut prouver le lemme suivant.

   > Soient t, f deux fonctions et soit $b \in \mathbb{N}^{\geq 2}$. Si t est éventuellement non-décroissante et f est $b-$lisse, et si $t \in \Omega(f|n \in P_b)$, alors $t \in \Omega(f)$.

   Prouvez-le.

4. (**20 points**) Soit $T(n)$ la récurrence donnée par $T(0) = 0$ et $T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + n$ pour $n > 0$. Trouvez la solution exacte pour $n \in P_2$ et l'ordre exacte pour $n \in \mathbb{N}$, si possible. N'oubliez pas de justifier vos dires.

5. (**20 points**) Donnez un exemple de deux fonctions f, g telles que $f \notin O(g)$ et $g \notin O(f)$.

6. (**20 points**) Prouvez que la relation $\equiv_\Theta$ sur $\mathcal{F} = \{f|\ f : \mathbb{N} \longrightarrow \mathbb{R}^{\geq 0}\}$ définie par $f \equiv_\Theta g$ si $f \in \Theta(g)$ est une relation d'équivalence.

7. (**20 points**) Utilisez les symboles $\subset$ et $=$ (mais **aucun** $\subseteq$) pour mettre en rang les ensembles ci-bas. Aucune justification n'est demandée.

$$O(2^{\sin n}) \qquad O(2^{\cos n}) \qquad O(7) \qquad O(2^{\log_3 n}) \qquad O(3^{\lg n}).$$

Réponse:

$O(\qquad\qquad) \qquad O(\qquad\qquad) \qquad O(\qquad\qquad) \qquad O(\qquad\qquad) \qquad O(\qquad\qquad).$

8. (**20 points**) Soit $G = (V, E)$ un graphe simple non-orienté avec n sommets et m arêtes et soit $u \in V$. Donnez un algorithme pour trouver les distances de u à tous les sommets de G. Quelle est la complexité de votre algorithme (non, "Dijkstra" n'est pas la bonne réponse, on peut faire bien bien simple) ?

9. (**20 points**) Prouvez que la complexité de tout algorithme qui trie n clés par comparaison est dans $\Omega(n \ln n)$.

10. (**10+10 points**) Soit $k - COL$ le problème de décision défini par

> Données: un graphe $G = (V, E)$ et un $k \in \mathbb{N}^{>0}$
> Question: est-ce qu'il existe une fonction $c : V \longrightarrow \{0, 1, \ldots, k-1\} = [k]$ telle que $c(u) \neq c(v)$ quand $uv \in E$?

(1) Supposons que l'on ait prouvé que $3 - COL$ est NP-complet (en fait, c'est vrai). Prouvez que $k - COL$ est NP-complet pour $k \geq 3$.

(2) Est-ce que $2 - COL$ est NP-complet?

11. (**20 points**) Soit un ensemble de n clés données dans un tableau $L[1..n]$. En utilisant des comparaisons, on veut trouver les deux plus grandes clés et on veut que le nombre de comparaisons soit strictement inférieur à $n + \lg n$. Comment? Prouver votre réponse.

12. (**20 points**) Vous avez une batterie de 5 tests à passer dans un cours. Si vous en réussissez la majorité, vous réussissez le cours, sinon... Quelle est la probabilité de votre réussite du cours si vous savez que la probabilité de réussir un des tests est (toujours) 0.7? Vous **devez** utiliser une méthode algorithmique vue en cours – une réponse, même correcte, ne vaut rien si elle est obtenue autrement.

BONNE
CHANCE

ANNEXE

(1) Soit R la récurrence homogène

$$\sum_{i=0}^{k} a_i t_{n-i} = 0$$

et soit $p(x)$ le polynôme caractéristique de R, avec les racines distinctes $r_1, \ldots, r_\ell$ de multiplicités respectives $m_1, \ldots, m_\ell$. Alors la solution générale de R est

$$\sum_{i=1}^{\ell} \sum_{j=0}^{m_i-1} c_{ij} n^j r_i^n.$$

(2) Soit R^* la récurrence non homogène

$$\sum_{i=0}^{k} a_i t_{n-i} = \sum_{i=1}^{s} b_i^n q_i(n)$$

avec $b > 0$ et $q_i(n)$ un polynôme en n de degré d_i, $i = 1, \ldots, s$. Soit $p^*(x)$ le polynôme caractéristique de R^*, avec les racines distinctes $r_1, \ldots, r_\ell$ de multiplicités respectives $m_1, \ldots, m_\ell$,

$$p^*(x) = p(x) \prod_{i=1}^{s} (x - b_i)^{d_i+1}.$$

Alors la solution générale de R est

$$\sum_{i=1}^{\ell} \sum_{j=0}^{m_i-1} c_{ij} n^j r_i^n.$$

(3) Soit $t(n)$ une fonction éventuellement non-décroissante donnée par la récurrence

$$t(n) = \ell t(\frac{n}{b}) + cn^k$$

pour $n \geq n_0$, avec $b, k, n_0 \in \mathbb{N},\ b \geq 2,\ n_0 \geq 1,\ n > 0,\ c \in \mathbb{R}^{>0}$. Alors

$$t(n) \in \begin{cases} \Theta(n^k) & \text{quand } b^k > \ell \\ \Theta(n^{\log_b \ell}) & \text{quand } b^k < \ell \\ \Theta(n^k \log n) & \text{quand } b^k = \ell \end{cases}$$

et les relations semblables sont vraies pour

$$t(n) = \ell t(\frac{n}{b}) + f(n),$$

$f(n) \in \Theta(n^k)$ et, *mutatis mutandis*, $f(n) \in O(n^k), f(n) \in \Omega(n^k)$.

Remarque. Quand on change la variable ou le codomaine, les conditions originales ne disparaissent pas mais sont traduites dans le nouveau paradigme.