Épreuve d'informatique théorique

Cette épreuve contient 20 questions réparties en 4 thèmes de 5 questions chacun :

1. La combinatoire;
2. La théorie des graphes;
3. La théorie des jeux et des énigmes;
4. Les probabilités.

La pondération est la même pour chacune des questions. Bonne chance !

Les questions suivantes portent sur la **combinatoire**, c'est-à-dire la théorie qui consiste à compter ou énumérer des objets.

1. On dit d'une chaîne binaire w qu'elle *évite* un motif u si elle ne s'écrit pas sous la forme $w = xuy$, où x et y sont des mots quelconques (possiblement vides).
   (a) Pour tout entier $n \geq 0$, soit $A(n)$ le nombre de chaînes binaires de longueur n qui évitent le motif 0. Donnez la valeur de $A(n)$ pour tout n.
   (b) De la même façon, soit $B(n)$ le nombre de chaînes binaires de longueur n qui évitent le motif 00. Donnez la valeur de $B(n)$ pour toute valeur de n.
   (c) Même question, mais avec $C(n)$ le nombre de chaînes binaires qui évitent le motif 000.

2. Un *arbre binaire* est une arborescence dont chaque sommet a au plus 2 enfants. Pour tout entier $n \geq 0$, soit $A(n)$ le nombre d'arbres de n noeuds.
   (a) Montrez que les premières valeurs de $A(n)$ sont bien celles illustrées dans le tableau ci-bas :

n	0	1	2	3	4
$A(n)$	1	1	2	5	14

   (b) Il est bien connu que $A(n)$ vérifie les équations suivantes :

$$\begin{aligned} A(0) &= 1, \\ A(n) &= \sum_{i=0}^{n-1} A(i)A(n-i-1). \end{aligned}$$

   Justifiez ces deux équations. *Indice :* Réfléchissez à la construction récursive des arbres binaires.

3. Combien existe-t-il de numéros de téléphone de 10 chiffres (on suppose que les 10^{10} possibilités sont toutes valides) qui contiennent au moins une fois chacun des chiffres impairs ? Par exemple, le numéro 418-937-5351 contient les chiffres 1, 3, 5, 7 et 9, de sorte qu'il devrait être compté.

4. Considérez une grille de r rangées et c colonnes (voir Figure 3). Dans cette question, on s'intéresse à compter le nombre de chemins à partir du point $(0, 0)$ jusqu'au point (c, r) n'utilisant que des déplacements vers la droite (identifiés par la lettre d) et vers le haut (identifiés par h). Donnez une formule qui compte le nombre de chemins distincts de $(0, 0)$ vers (c, r), pour toutes valeurs $r, c \geq 1$.

5. On dit d'un nombre entier qu'il est *répétitif* si tous ses chiffres en base 10 sont identiques. Par exemple, 33 et 999 sont des nombres répétitifs. Aussi, soit E un ensemble de 12 nombres distincts quelconques entre 1 et 99 inclusivement. Montrez qu'il existe au moins deux nombres $x, y \in E$ tels que $x - y$ est un nombre répétitif.

Les questions suivantes portent sur la **théorie des graphes**. Un *graphe simple* est un couple $G = (V, E)$ où V est son ensemble de *sommets* et $E \subseteq \mathcal{P}_2(V)$ est son ensemble d'*arêtes*, où $\mathcal{P}_2(V)$ dénote l'ensemble des paires d'éléments distincts de V. Par exemple, le graphe $G = (V, E)$ défini par les ensembles

$$V = \{a, b, c, d\} \quad \text{et} \quad E = \{\{a, b\}, \{a, c\}, \{b, c\}, \{c, d\}\}$$

est représenté à la figure 2. À noter que dans un graphe simple, il n'y a pas d'arête multiple ni de boucle (une arête d'un sommet vers lui-même).

6. Certaines familles de graphes simples sont souvent utilisées. En voici quelques-unes (voir figure 1) :

   (i) Pour tout entier $n \geq 1$, le graphe *complet* de n sommets K_n;

   (ii) Pour tout entier $n \geq 3$, le *cycle* de n sommets C_n;

   (iii) Pour tout entier $n \geq 3$, la *roue* de $n + 1$ sommets W_n;

   (iv) Pour tout entier $n \geq 1$, l'*hypercube* de dimension n est noté H_n;

   (v) Pour toute paire d'entiers $m, n \geq 1$, le graphe *biparti complet* $K_{m,n}$;

   Pour toutes valeurs n et m bien définies, donnez le nombre d'arêtes des graphes suivants :

   (a) K_n;

   (b) C_n;

   (c) W_n;

   (d) H_n;

   (e) $K_{m,n}$.

7. Soit $G = (V, E)$ un graphe simple. On dit d'un ensemble de sommets $U \subseteq V$ qu'il est un *transversal d'arêtes* si chaque arête de G a au moins une de ses extrémités dans U, c'est-à-dire que pour toute arête $\{u, v\} \in E$, on a $u \in U$ ou $v \in U$. Par exemple, dans le graphe de la figure 2, l'ensemble $\{a, c\}$ est un transversal d'arêtes. Un transversal d'arête U est dit *minimum* pour G s'il n'existe aucun transversal d'arêtes U' de plus petite cardinalité, c'est-à-dire tel que $|U'| < |U|$.

   Pour toutes valeurs de n et m bien définies, donnez la taille d'un transversal d'arêtes minimum des graphes suivants :

   (a) K_n;

   (b) C_n;

   (c) W_n;

   (d) H_n;

   (e) $K_{m,n}$.

8. On dit d'un graphe G qu'il est *hamiltonien* s'il admet au moins un cycle passant par chaque sommet **exactement** une fois.

   Soit HAMD le problème de *décider* si un graphe simple G est hamiltonien ou non et soit HAM le problème de *trouver* un cycle hamiltonien dans un graphe G (on retourne la valeur *rien* si un tel cycle n'existe pas). Autrement dit, on considère les deux fonctions suivantes :

   - EstHamiltonien(G : graphe) : booléen;
   - CycleHamiltonien(G : graphe) : cycle ou *rien*.

   Montrez que les deux problèmes sont de même difficulté à complexité polynomiale près. En d'autres termes, si EstHamiltonien utilise un temps constant, alors on peut implémenter CycleHamiltonien de telle sorte qu'elle utilise un temps polynomial (et vice-versa). *Remarque :* Vous devez écrire du pseudocode implémentant chacune des deux fonctions par rapport à l'autre.

9. Étant donnés un graphe non orienté $G = (V, E)$ et un entier positif k, on dit d'une fonction $c : V \to \{1, 2, \ldots, k\}$ qu'elle est un *k-coloriage* de G si pour toute arête $\{u, v\} \in E$, on a $c(u) \neq c(v)$, c'est-à-dire que deux sommets adjacents ont une couleur différente. Si G peut être colorié à l'aide de k couleurs, alors on dit qu'il est k-coloriable.

   D'autre part, on dit que G est k-parti s'il existe des ensembles $V_1, V_2, \ldots, V_k$ tels que

   1. $V_1 \cup V_2 \cup \cdots \cup V_k = V$,
   2. $V_i \cap V_j = \emptyset$ et
   3. Pour tout arête $\{u, v\}$, si $u \in V_i$ et $v \in V_j$, alors $i = j$.

   Autrement dit, on peut partitionner l'ensemble de sommets V en k morceaux de sorte que les arêtes reliant deux sommets dans un même morceau sont interdites.

   Montrez que tout graphe k-parti est k-coloriable.

10. Soit k-COLORD le problème de *décider* si un graphe non orienté G est k-coloriable et k-COLOR le problème de *trouver* un k-coloriage de G. Montrez que les problèmes k-COLORD et k-COLOR sont de même difficulté à complexité polynomiale près. *Indice :* La question 9 devrait vous être utile.

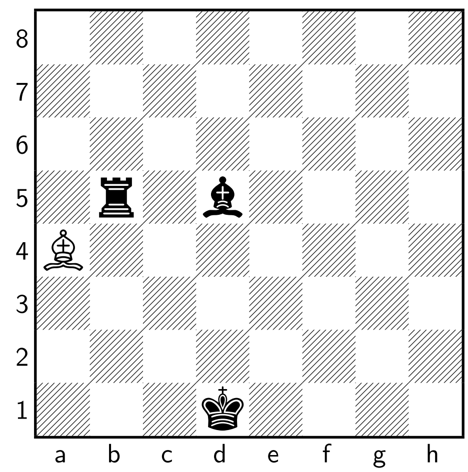
Les questions suivantes portent sur la **théorie des jeux et des énigmes**.

11. Cette question porte sur le célèbre *jeu de Nim*, aussi appelé *jeu des allumettes.* Il s'agit d'un jeu à deux joueurs, appelés joueur A et joueur B, qui, à tour de rôle, doivent retirer 2, 3 ou 4 allumettes d'un tas d'allumettes. Initialement, on suppose qu'il y a n allumettes, où $n \geq 1$. Le gagnant est celui qui retire la ou les dernières allumettes.

    On dit d'un joueur qu'il a une *stratégie gagnante* s'il peut gagner à tous les coups, peu importe ce que l'autre joueur décide de faire.

Expliquez sous quelle condition le joueur A ou le joueur B a une stratégie gagnante au jeu de Nim. *Remarque :* Dans le jeu de Nim, lorsque la valeur de n est fixée, un des deux joueurs a nécessairement une stratégie gagnante.

12. (Tiré du livre *The Chess Mysteries of the Arabian Knights*, de R. Smullyan) Considérez la position suivante aux échecs.



La position du roi blanc a été cachée. De plus, on ne sait pas si c'est au tour de blanc ou noir de jouer.

Devinez quelle est la position du roi blanc, sachant que tous les coups précédents ont été légaux et qu'il n'y a qu'une solution possible.

13. Nous nous intéressons maintenant à une variante du jeu d'échecs traditionnels, dans laquelle chaque joueur peut jouer *deux coups consécutifs*. Sachant que ce sont les blancs qui commencent et les noirs qui jouent en deuxième, montrez qu'avec cette variante, les noirs ne peuvent pas avoir une stratégie gagnante. *Indice :* procédez par l'absurde.

14. Vous vous trouvez dans un édifice de 100 étages (numérotées de 1 à 100) avec 2 oeufs. Chacun de ces oeufs a la particularité qu'il existe un nombre n entre 1 et 100 (le nombre n est le même pour les 2 oeufs) tel que :

    - si vous le laissez tomber d'un étage entre 1 et n, alors il ne se casse pas;
    - si vous le laissez tomber d'un étage entre $n + 1$ et 100, alors il se casse.

    Proposez une stratégie optimale pour deviner la valeur de n qui minimise le nombre de fois que vous devez laisser tomber un oeuf. Justifiez votre réponse.

15. Vous êtes un homme riche et sans scrupule, ayant en votre possession 10 esclaves. Dans six heures aura lieu un banquet auquel sont conviés de nombreux invités. Cependant, vous venez d'apprendre que, parmi les 1000 tonneaux de vin que vous avez l'intention de servir, exactement un d'entre eux a été empoisonné. De plus, vous savez que les effets du poison, lorsque ingéré, ne se manifestent qu'entre 30 minutes et 5 heures, selon la personne. Vous décidez donc de tester les tonneaux de vins sur vos esclaves.

Proposez une stratégie qui vous permettra de déterminer avec certitude quel tonneau de vin est empoisonné. Notez qu'un esclave peut goûter à autant de tonneau de vin que nécessaire.

Note: Bien qu'il soit plus humain d'annuler simplement le banquet, cette réponse ne vous donnera aucun point...

Les questions suivantes portent sur la **théorie des probabilités**.

16. Considérez une fonction NonUniforme() qui retourne au hasard la valeur 0 avec probabilité 0.7 et la valeur 1 avec probabilité 0.3. On suppose que c'est la seule fonction que vous pouvez utiliser pour simuler le hasard, les autres fonctions habituelles n'étant pas disponibles. Évidemment, vous pouvez appeler la fonction NonUniforme() autant de fois que vous le désirez. En utilisant la fonction NonUniforme(), donnez le pseudocode d'une fonction Uniforme() qui retourne la valeur 0 ou 1 avec probabilité uniforme.

17. Supposez que pour un problème de décision donné P, vous ayez à votre disposition deux algorithmes, appelés A et B. Vous avez également les informations suivantes :

    - Lorsque l'algorithme A retourne *vrai*, vous savez que la réponse au problème P est nécessairement *vrai*. En revanche, si l'algorithme A retourne *faux*, il peut se tromper avec probabilité 0.05.
    - Lorsque l'algorithme B retourne *faux*, vous savez que la réponse au problème P est nécessairement *faux*. En revanche, si l'algorithme B retourne *vrai*, il peut se tromper avec probabilité 0.1.

    Est-il possible de construire un algorithme C à partir des algorithmes A et B de telle sorte que l'algorithme C retourne *vrai* ou *faux* sans jamais se tromper ? Justifiez.

18. Considérez un arbre binaire de recherche quelconque. On suppose que les services suivants sont disponibles pour tout arbre T :

    - T.estVide() retourne vrai si et seulement si T est un arbre vide;
    - T.contenu() retourne la valeur contenue dans la racine de l'arbre T;
    - T.taille() retourne le nombre total de noeuds dans l'arbre T;
    - T.gauche() retourne le sous-arbre gauche de T (possiblement vide);
    - T.droit() retourne le sous-arbre droit de T (possiblement vide).

    En n'utilisant que les services ci-haut, donnez le pseudocode d'une fonction

    ValeurAleatoire(T : arbre binaire de recherche)

    qui retourne au hasard une valeur présente dans l'arbre T (on suppose qu'une valeur ne peut pas apparaître plus d'une fois, pour simplifier). Notez que chaque valeur doit avoir la **même probabilité** d'être retournée. Aussi, vous pouvez supposer qu'il existe une fonction random(n) qui retourne un nombre entier entre 0 et $n-1$ (inclusivement) au hasard avec probabilité uniforme.

19. Considérez un ensemble de n entiers quelconques (distincts). On souhaite former un échantillon de i entiers distincts ($1 \leq i \leq n$) tirés aléatoirement parmi les n entiers disponibles (c'est donc un tirage sans remise). Pour cela, on utilise l'algorithme suivant :

```
fonction ÉCHANTILLON(E : ensemble d'entiers, i : entier)
    S ← ∅
    tant que S.TAILLE ≠ i faire
        e ← un élément choisi aléatoirement dans E
        si e ∉ S alors
            S.INSÉRER(e)
        fin si
    fin tant que
    retourner S
fin fonction
```

   (a) Quelle est la probabilité que l'algorithme ne termine pas ?

   (b) Montrez qu'en moyenne, le nombre de tours de boucle **tant que** effectués est $\mathcal{O}(n \log n)$, où n est la taille de l'ensemble E et $0 \leq i \leq n$.

20. Considérez un tableau T de n entiers non ordonnés et pas nécessairement distincts. Soit x un entier apparaissant k fois dans le tableau T. Clairement, $0 \leq k \leq n$.

   On s'intéresse aux nombres de valeurs inspectées lorsqu'on effectue une fouille linéaire du tableau T. Plus précisément, on considère l'algorithme suivant (T est indicé à partir de 0) :

```
fonction ESTDANSTABLEAU(T : tableau, x : élément)
    Soit n le nombre d'éléments dans T
    i ← 0
    tant que i < n et T[i] ≠ x faire
        i ← i + 1
    fin tant que
    retourner i < n
fin fonction
```

   Pour toutes valeurs de k possibles, donnez une formule indiquant le nombre de valeurs moyennes qu'on s'attend à inspecter? Vous devez donner une expression **exacte**, c'est-à-dire pas avec la notation $\mathcal{O}$. *Indice :* Considérez séparément les cas $k = 0$, $k = 1$ et $k > 1$.

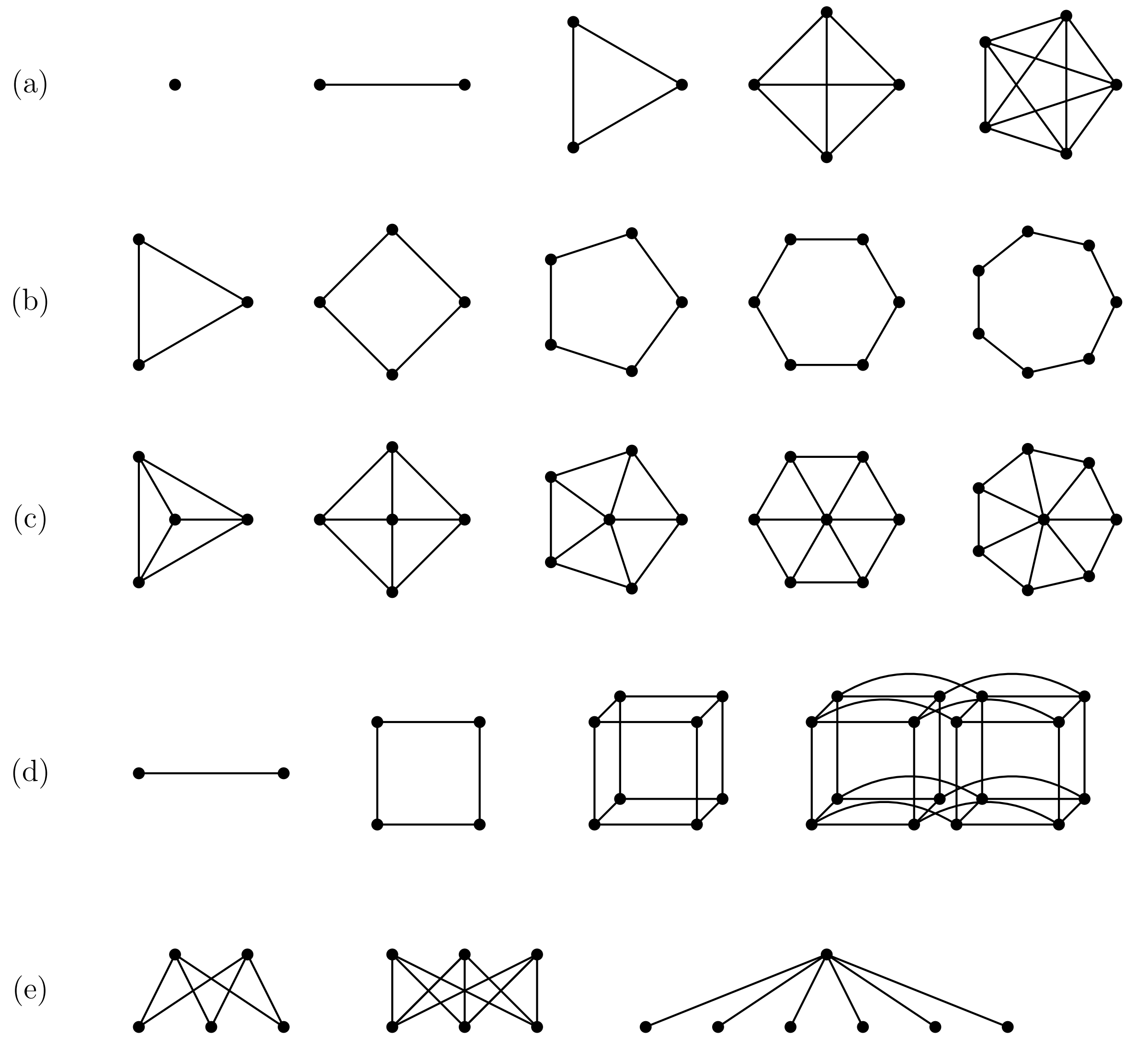


Figure 1: Quelques familles de graphes célèbres. (a) Le graphe complet K_n pour $n = 1, 2, 3, 4, 5$. (b) Le cycle C_n pour $n = 3, 4, 5, 6, 7$. (c) La roue W_n pour $n = 3, 4, 5, 6, 7$. (d) L'hypercube H_n pour $n = 1, 2, 3, 4$. (e) Les graphes bipartis $K_{2,3}$, $K_{3,3}$ et $K_{1,6}$.

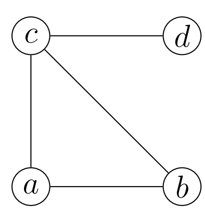


Figure 2: Le graphe $G = (V, E)$, où $V = \{a, b, c, d\}$ et $E = \{\{a, b\}, \{a, c\}, \{b, c\}, \{c, d\}\}$.

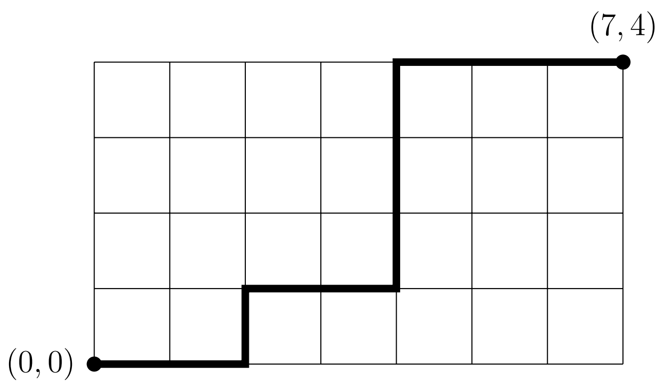


Figure 3: Une grille avec $r = 4$ et $c = 7$. Le chemin *ddhddhhhddd* est représenté par un trait gras.