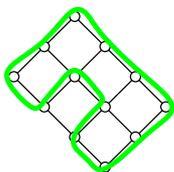


Indiquez votre numéro d'anonymat : No :

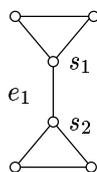
Question 1. Cocher les affirmations qui sont correctes :

- Tout graphe fortement connexe possède un cycle passant par tous les sommets.
- Tout graphe qui possède un cycle est connexe.
- Pour tout graphe G sans cycle, on a $|E(G)| = |V(G)| - 1$.
- Pour tout arbre A , on a $|E(A)| = |V(A)| - 1$.
- Le graphe G_2 dessiné ci-dessous est Eulérien.
- Le graphe G_3 dessiné ci-dessous est Eulérien.
- Le graphe G_1 dessiné ci-dessous possède 4 sommets de degré 2, 6 de degré 3 et 2 de degré 4, donc $|E(G_1)| = 4 \times 2 + 6 \times 3 + 2 \times 4 = 34$.

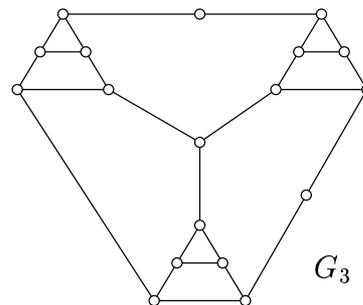
Question 2. Un cycle *Hamiltonien* dans un graphe G est une chaîne $s_1, e_1, s_2, e_2, \dots, s_n, e_n, s_{n+1}$ telle que $s_{n+1} = s_1$ et $|\{s_1, \dots, s_n, s_{n+1}\}| = |V(G)| = n$. Dit autrement, c'est un cycle passant une et une seule fois par chacun des sommets du graphe.



G_1



G_2



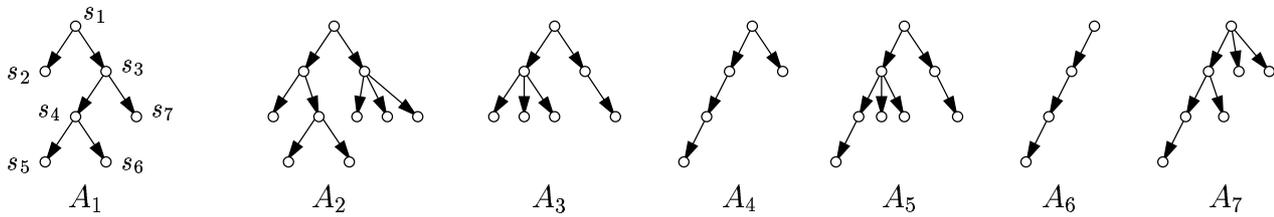
G_3

Dans l'exemple ci-dessus, G_1 possède un cycle Hamiltonien (dessiné en gras). Mais G_2 ne possède pas de cycle Hamiltonien car tout cycle contenant e_1 doit contenir s_1 et s_2 au moins deux fois, ce qui n'est pas possible pour un cycle Hamiltonien.

Est-ce que le graphe G_3 possède un cycle Hamiltonien ? Si oui, dessinez directement sur G_3 un cycle Hamiltonien, sinon justifier en quelques mots.

Justification :

Question 3. On rappelle que dans un graphe orienté, un *chemin* est une suite d'arcs a_1, a_2, \dots, a_k telle que $H(a_i) = T(a_{i+1})$ pour tout $i = 1, \dots, k - 1$. Le sommet $T(a_1)$ est appelé l'*origine* du chemin, $H(a_k)$ l'*extrémité*. Toute la suite de l'exercice concerne les arbres, comme les sept graphes dessinés ci-après.



La *hauteur* d'un sommet s , notée $h(s)$, est le nombre d'arcs du plus long chemin dont l'origine est le sommet s . Dans l'arbre A_1 dessiné ci-dessus, $h(s_1) = 3$ et $h(s_2) = 0$. Le *poids* d'un sommet s , noté $p(s)$, est le nombre de sommets qui sont extrémité d'un chemin d'origine s . Dans l'arbre A_1 , $p(s_3) = 4$ et $p(s_2) = 0$.

Calculer la hauteur et le poids de chaque sommets de l'arbre A_1 en complétant le tableau ci-dessous :

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
h	3	0					
p		0	4				

Un arbre est *h-équilibré* si pour tout arc a , $h(T(a)) - h(H(a)) \leq 2$. Un arbre est *p-équilibré* si pour tout arc a , $p(H(a)) \leq \frac{1}{2}p(T(a))$. A_1 n'est pas h-équilibré car $h(s_1) - h(s_2) = 3$ alors qu'il y a un arc de s_1 à s_2 . A_1 n'est pas p-équilibré car $p(s_3) > \frac{1}{2}p(s_1)$ alors qu'il y a un arc de s_1 à s_3 .

Pour les arbres A_2, \dots, A_7 dessinés ci-dessus, complétez le tableau suivant par « oui » ou « non » :

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
h-équilibré?	non						
p-équilibré?	non						

Question 4. Un programmeur médiocre vient d'écrire un programme censé calculer certaines statistiques sur un graphe quelconque. La complexité du programme peut être exprimée sous la forme $O(|V|^2 + |E|)$. Il teste son programme sur plusieurs graphes avec les résultats suivants :

	$ V $	$ E $	temps d'exécution
G_4	100	0	2 s
G_5	100	4000	4 s

Quel temps d'exécution approximatif peut on espérer sur un graphe avec $|V| = 10\,000$ et $|E| = 20\,000$? Cocher la bonne case et justifier en quelques mots.

- quelques minutes quelques heures quelques jours
 quelques mois quelques années plus de 10 ans

Justification :

Question 5. Expliquer en une phrase ce que fait la fonction Python `mystere` :

```
def mystere (n):
    result = 0
    while n > 0:
        if n % 2 == 1:
            result = resultat + 1
        n = n / 2
    return result
```

Explication :

FIN.