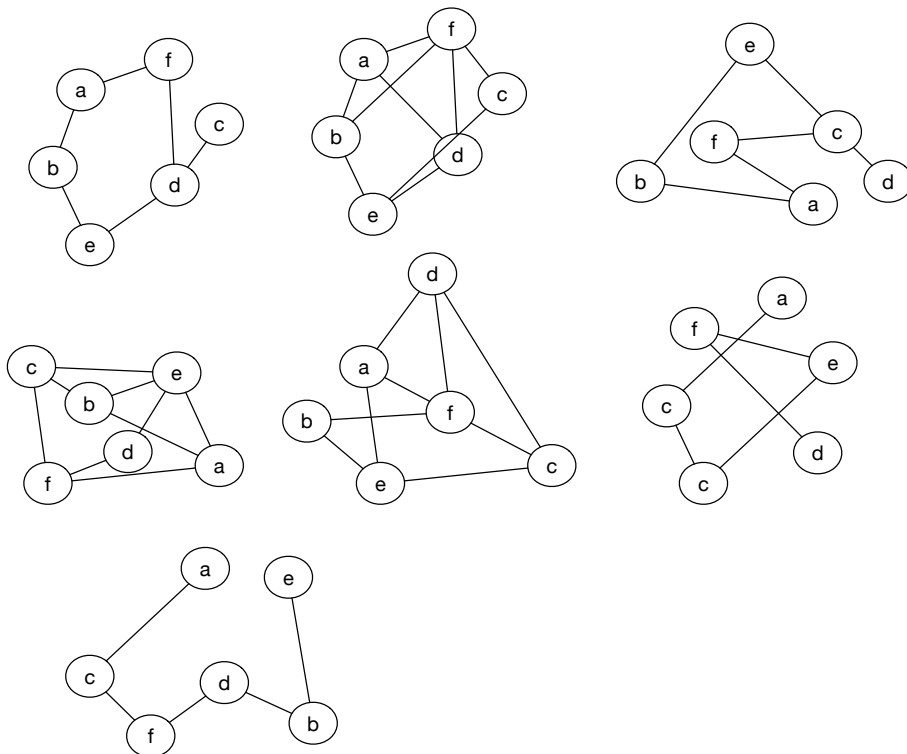


Modalités : La durée de l'épreuve est de 2h, notée sur 23 points. Une feuille A4 recto-verso de notes de cours est autorisée. Justifiez vos réponses avec des illustrations si nécessaire. Les questions peuvent être traitées dans le désordre, et ne sont pas classées par ordre de difficulté.

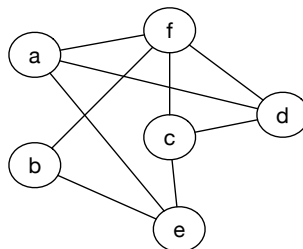
Exercice 1: Compréhension du cours (5 pts)

1-a) Pouvez vous donner les propriétés de la relation gRg' , vraie si le graphe g est isomorphe à g' ?

Vous classerez ensuite les graphes suivants en groupes (chaque groupe ne comportant que des graphes homomorphes).



1-b) Le graphe suivant est-il planaire ? Justifiez ? Combien de couleurs sont nécessaires pour le colorer ?



1-c) Dessinez une arborescence binaire équilibrée avec les éléments suivants $X = \{25, 15, 12, 19, 37, 3, 78, 12\}$. Vous proposerez un algorithme permettant de retourner si une valeur est présente dans un tel arbre. Vous en spécifierez sa complexité.

1-d) Soit P l'ensemble d'arêtes (minimum) définissant une face dans un graphe planaire.

Théorème 1 *Tout graphe planaire $G(S,A)$ topologique connexe et dont la plus petite face est P contient au plus $\frac{|P|}{|P|-2}(|S| - 2)$ arêtes*

En vous appuyant sur la formule d'Euler*, pouvez vous démontrer une telle propriété ?

Exercice 2: Codage matriciel (3 pts)

- 2-a) Rappelez à quoi correspond le codage sous forme de matrice d'incidence pour un graphe G
- 2-b) Rappelez ce qu'est un linegraphe pour le graphe non orienté $G(S,A)$
- 2-c) Donnez un algorithme qui permet d'obtenir la matrice d'incidence d'un linegraph avec la matrice d'incidence du graphe d'origine.

Exercice 3: Time evolving graph (5 pts)

Soit un essaim de robots mobiles communicants, modélisé sous la forme d'un time-evolving graphe. Ainsi, une arête existe dans le graphe entre deux robots s'ils sont suffisamment proches l'un de l'autre. Comme ils sont mobiles, une arête n'a d'existence que pendant des intervalles de temps précis, que nous supposons ici connus à l'avance (ex : une communication entre le robot a et b est possible de t_0 à t_1 puis de t_2 à t_3 , etc.) Nous supposons que les communications sont bidirectionnelles : si l'arête existe entre les sommets u et v , alors u peut envoyer des données à v et v à u .

Nous voulons ici étudier la transmission d'un paquet d'un mobile s à un mobile d , éventuellement stocké puis relayé par des mobiles intermédiaires. On néglige le temps de transmission vis-à-vis du temps d'existence d'une arête.

- 3-a) Vous discuterez de la notion de plus court chemin et de connexité (faible et forte) par rapport à un graphe classique. Vous pourrez notamment discuter des métriques possibles pour la notion de "longueur" d'un chemin.

Vous donnerez un exemple et un codage possible d'un time evolving graph.

- 3-b) Soit la relation uRv ssi u possède un chemin vers v . Pouvez vous donner les propriétés de R ?
 - 3-c) Dijkstra est-il adaptable pour calculer un plus court chemin (en heure d'arrivée) du sommet s au sommet d (en sachant que l'instant de génération du paquet est connue t_0) ? Existe-t-il des conditions à respecter ?
 - 3-d) Vous en donnerez une version en pseudo-code (vous vous focaliserez sur les différences avec le Dijkstra d'origine).
 - 3-e) Peut-on avec un tel algorithme trouver toutes les routes **vers** un sommet d ? Pourquoi ?
- question bonus** : comment pourriez vous faire (autrement qu'en testant tous les départs possibles) ?

*. $n - a + f = 2$

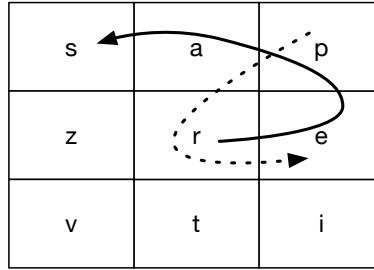


FIGURE 1 – Exemple d’une instance d’un boggle de 9 dés, avec deux mots possibles

Exercice 4: Jeu du Boggle (5 pts)

Soit un damier carré de $N \cdot N^\dagger$ lettres, et un dictionnaire comprenant tous les mots existant de la langue française. Nous supposons que nous pouvons partir de n’importe quelle lettre et déplacer le pointeur horizontalement, verticalement ou en diagonal. Dans l’exemple figure 1, le boggle peut nous donner les mots "pré" ou "repas". Dans le jeu original, il n’est pas possible d’utiliser pour un mot plusieurs fois le même dé.

- 4-a) Pouvez vous donner une représentation sous forme de graphe de ce Boggle ?
- 4-b) Quel est l’algorithme que vous proposez pour trouver tous les mots possibles ?
- 4-c) Proposez une façon d’organiser le dictionnaire pour accélérer votre recherche
- 4-d) Comment modifier votre algorithme pour pouvoir utiliser plusieurs fois le même dé (et rendre par exemple possible le mot "été" dans le boggle de la figure 1 ?

Exercice 5: Connexité (5 points)

Soit G un graphe simple ayant n sommets et $n-1$ arêtes qui n’est pas un arbre. On suppose qu’un sommet isolé est un arbre "trivial" (une racine singleton sans arête).

1. Prouvez que G n’est pas connexe.
2. Prouvez que G possède au moins une composante connexe qui est un arbre.
3. Prouvez que G possède au moins une composante connexe qui n’est pas un arbre.
4. Prouvez que si G possède exactement deux composantes connexes, alors celle qui n’est pas un arbre possède exactement un cycle.

[†]. N étant typiquement un entier de 3 à 5