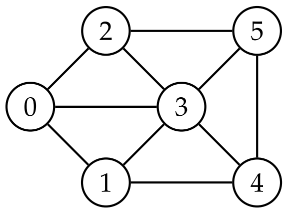
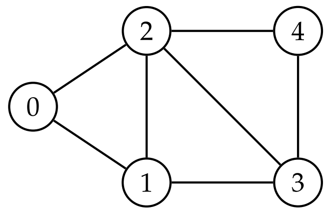
TD10 : Annale d'examen

Chaque réponse doit être soigneusement justifiée. La qualité de la rédaction est un critère primordial dans la notation.

Marche aléatoire pour la coloration de graphe.
Une k-coloration d'un graphe G est l'attribution à chaque sommet de G d'une couleur parmi un ensemble de k couleurs possibles. Une coloration est *valide* si deux sommets reliés par une arête ont des couleurs distinctes. Un graphe qui admet une k-coloration valide est dit *k-coloriable*. On s'intéressse dans ce problème au calcul d'une 3-coloration par un algorithme probabiliste.

1. Les deux graphes ci-dessous sont-ils 3-coloriables ? Le cas échéant, donner une 3-coloration valide.

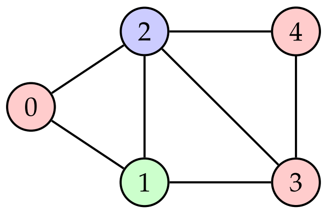


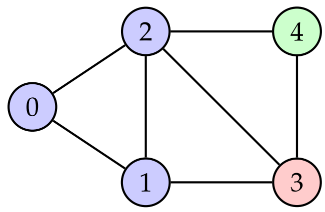


2. Un premier algorithme probabiliste pour la 3-coloration consiste à tirer aléatoirement et indépendamment une couleur (parmi les trois disponibles) pour chaque sommet.

- **i.** On fixe une arête $e = \{u, v\}$ dans G. Quelle est la probabilité que cet arête soit *valide*, c'est-à-dire que ses deux sommets aient une couleur différente ?
- **ii.** En déduire que la probabilité d'échec de l'algorithme est bornée par $m/3$ où m est le nombre d'arêtes de G.
- **iii.** Pourquoi cet algorithme n'est pas un bon algorithme ?

On va maintenant utiliser la technique de la marche aléatoire pour trouver une 3-coloration d'un graphe 3-coloriable. Pour cela, on définit la *distance* entre deux colorations d'un même graphe G comme le nombre de sommets qui n'ont pas la même couleur dans les deux colorations. Par exemple, les deux colorations ci-dessous sont à distance 3.





L'idée de l'algorithme est de partir d'une coloration aléatoire du graphe, et de la modifier localement tant qu'il existe une arête non valide dans le graphe en rendant une de ces arêtes valides.

3. Écrire le pseudo-code d'un algorithme probabiliste basé sur l'idée ci-dessus.

Pour étudier la complexité de cet algorithme, on fixe une 3-coloration du graphe et on étudie l'évolution de la distance d entre la coloration courante du graphe et cette coloration fixée. On note Δd l'évolution de cette distance lors d'une étape de l'algorithme, c'est-à-dire que la distance à la coloration fixée passe de d à $d + \Delta d$ quand on effectue une étape de l'algorithme.

4. Montrer que $\Delta d \in \{-1, 0, 1\}$.

On peut distinguer deux cas lors d'une étape de l'algorithme : soit aucun des deux sommets de l'arête rendue valide n'avait la *bonne couleur* (la même couleur que dans la coloration fixée), soit l'un des deux sommets avait la bonne couleur.

5. Calculer les probabilités que $\Delta d = -1$, $\Delta d = 0$ et $\Delta d = 1$, pour $1 \leq d < n$, dans ces deux cas. Quel est le pire cas pour l'algorithme ?

On note $p(d)$ la probabilité que l'algorithme atteigne la coloration fixée, en partant d'une coloration à distance d. On suppose qu'à chaque étape de l'algorithme, on est dans le pire cas de la question précédente.

6. Montrer que $p(d-1) - 3p(d) + 2p(d+1) = 0$ pour $1 \leq d < n$ où n est le nombre de sommets du graphe. Donner les équations des cas limites $p(0)$ et $p(n)$.

On admet que l'unique solution à l'équation précédente vérifie $p(d) \geq 2^{-d}$.

7. Combien y a-t-il de colorations de G à distance d de la coloration fixée ?
8. En déduire que la probabilité de succès de l'algorithme est $\geq (2/3)^n$, et calculer l'espérance du nombre de fois qu'il faut exécuter l'algorithme avant de trouver une 3-coloration valide.
9. On ne peut pas *a priori* savoir si une exécution va s'arrêter (c'est-à-dire terminer sur une 3-coloration valide). Donner une ou plusieurs stratégies possibles pour obtenir à partir des idées précédentes un algorithme de 3-coloration de graphe de type Las Vegas.