CORRIGÉ

Tle

Corrigé détaillé
4 exercices

Graphes — Structures et algorithmes de parcours

> Ce qui est noté au contrôle, ce n'est pas le résultat seul : c'est le raisonnement posé, puis le calcul mené jusqu'au bout avec l'unité.

1 Vocabulaire. Répondre en une phrase précise.

1. a. Le graphe **orienté** a des relations non réciproques — des **arcs**. La **connexité** est la possibilité d'aller de tout sommet à tout autre.
2. b. La **matrice d'adjacence** teste l'adjacence en O(1) mais occupe n^2 ; la **liste d'adjacence** occupe un espace proportionnel au nombre d'arêtes.
3. c. Le **BFS** une **file** ; le **DFS** une **pile** ou la récursivité.
4. d. Le **BFS**, à condition que le graphe soit **non pondéré** — ou que toutes les arêtes coûtent la même chose.
5. e. Le plus court chemin dans un graphe **pondéré**, à condition que les poids soient **positifs**.

2 Un développeur cherche la plus courte distance entre deux sommets d'un réseau non pondéré et emploie un parcours en profondeur. Analyser.

1. a. **Oui**, si les deux sommets sont dans la même composante connexe. Le **DFS** visite tous les sommets atteignables.
2. b. **Non**, sauf par chance. Le **DFS** s'engage dans la **première direction disponible** et peut faire un immense détour avant d'atteindre un sommet pourtant voisin du départ.
3. c. Parce qu'il explore **couche par couche** : tous les sommets à une arête, puis à deux, et ainsi de suite. Le **premier moment** où il atteint un sommet correspond donc nécessairement au **plus petit nombre d'arêtes**.
4. d. Remplacer la **pile** par une **file**. Le reste de l'algorithme est identique — c'est la seule différence entre les deux parcours.
5. e. Parce que sur un **petit graphe**, ou sur un graphe où l'ordre des voisins est favorable, le **DFS** peut renvoyer le bon résultat. Le test passe, et l'erreur n'apparaît que sur certaines topologies.

3 Un réseau social compte 1 000 000 de comptes, chacun ayant en moyenne 200 relations. Raisonner sur la représentation.

1. a. Une matrice compte n^2 cases :
$$1000000 \times 1000000 = 1000000000000$$
2. Soit **mille milliards** de cases — hors de portée.
3. b. Chaque compte a 200 relations, et chaque arête est comptée deux fois :
$$1000000 \times 200 \div 2 = 100000000$$
4. Soit **cent millions** d'arêtes.
5. c. Extrêmement **creux** : cent millions d'arêtes pour mille milliards de cases possibles, soit un dix-millième.
$$1000000000000 \div 100000000 = 10000$$
6. d. La **liste d'adjacence** : son espace est proportionnel au **nombre d'arêtes**, donc de l'ordre de cent millions et non de mille milliards. Une matrice stockerait presque uniquement des zéros.

4 Parcours et conditions. Raisonner.

1. a. Un **cycle** provoque une **boucle infinie** : l'algorithme revient indéfiniment sur des sommets déjà traités.
2. b. En mémorisant le **prédécesseur** de chaque sommet au moment où on l'atteint, puis en remontant depuis l'arrivée jusqu'au départ.
3. c. Parce que si **toutes les arêtes valent 1**, les deux algorithmes traitent les sommets dans le **même ordre** : le sommet non traité le plus proche est alors le suivant dans la file.
4. d. Parce qu'il **fige** la distance d'un sommet dès qu'il le traite, en supposant qu'aucun chemin ultérieur ne fera mieux. Un poids **négatif** rend cette hypothèse fausse, et le résultat est erroné **sans être signalé**.

Corrige-toi toi-même. Compte tes points exercice par exercice : c'est ainsi qu'on voit où l'on perd, et ce n'est presque jamais là où on le croit.

Ce qui est évalué	Points	Ce qui fait perdre le point
Choisir la représentation selon la densité	4 pts	Employer une matrice sur un graphe creux

Associer BFS à la file et DFS à la pile	4 pts	Inverser les deux structures
Réserver le plus court chemin au BFS	5 pts	Employer un DFS pour une distance
Marquer les sommets visités	3 pts	Boucler sur un cycle
Énoncer la condition de Dijkstra	4 pts	L'appliquer à des poids négatifs