

Graphes — Structures et algorithmes de parcours

NSI · Terminale
Contrôle n°1 · 60 min · sur 20

Exercice 1 — Représenter un graphe

5 pts 14 min

Choisir une représentation

Réponds à chaque question.

LA RÈGLE DU COURS

Un graphe a son vocabulaire, et deux représentations se partagent les usages selon une propriété du graphe.

1. Qu'est-ce qu'un graphe, et quand est-il orienté ? (1 pt)

Un ensemble de sommets reliés par des arêtes ; il est orienté quand la relation n'est pas réciproque.

→ On parle alors d'arcs plutôt que d'arêtes.

2. Que signifient connexité et cycle ? (1 pt)

La connexité demande si l'on peut aller de tout sommet à tout autre ; un cycle est un chemin qui revient à son point de départ.

→ Ce sont les deux propriétés de base à tester sur un graphe.

3. Quels sont l'avantage et le coût d'une matrice d'adjacence ? (1 pt)

Le test d'adjacence est en $O(1)$, mais l'espace occupé est en n^2 même si le graphe a peu d'arêtes.→ La case (i, j) indique s'il existe une arête entre les deux sommets.

4. Quels sont l'avantage et le coût d'une liste d'adjacence ? (1 pt)

L'espace est proportionnel au nombre d'arêtes, mais le test d'adjacence coûte davantage.

→ Elle associe à chaque sommet la liste de ses voisins.

5. Comment représenter un réseau social où chacun connaît une poignée de personnes sur des millions ? (1 pt)

Par une liste d'adjacence, le graphe étant extrêmement creux.

→ Une matrice y gaspillerait un espace considérable pour stocker presque uniquement des zéros.

Exercice 2 — Les deux parcours

4 pts 11 min

Distinguer BFS et DFS

Réponds à chaque question.

LA RÈGLE DU COURS

Deux parcours visitent les mêmes sommets et ne diffèrent que par une chose. Une précaution est commune aux deux.

1. Quelle structure le BFS emploie-t-il, et comment explore-t-il ? (1 pt)

Une file : il explore les voisins immédiats, puis leurs voisins, couche par couche.

→ C'est le parcours en largeur : la file impose l'ordre d'arrivée.

2. Quelle structure le DFS emploie-t-il, et comment explore-t-il ? (1 pt)

Une pile, ou la récursivité : il s'enfonce aussi loin que possible avant de revenir en arrière.

→ C'est le parcours en profondeur.

3. Qu'est-ce qui différencie les deux parcours, et qu'est-ce qui ne les différencie pas ? (1 pt)

Seule la structure qui mémorise les sommets à traiter les différencie ; tous deux visitent tous les sommets atteignables.

→ Le résultat en termes de sommets visités est le même.

4. Pourquoi faut-il marquer les sommets visités ? (1 pt)

Parce qu'un cycle provoquerait sinon une boucle infinie.

→ La précaution vaut pour les deux parcours.

Exercice 3 — La garantie du BFS

4 pts 11 min

Analyser la garantie du parcours en largeur

Réponds à chaque question.

LA RÈGLE DU COURS

Un seul des deux parcours donne au passage le plus court chemin, et cette garantie a une limite.

1. Pourquoi le DFS ne donne-t-il pas le plus court chemin ? (1 pt)

Parce qu'il trouve un chemin sans garantie sur sa longueur : il s'engage dans la première direction disponible.

→ Il peut faire un immense détour avant d'atteindre un sommet voisin du départ.

2. Pourquoi le BFS donne-t-il le plus court chemin ? (1 pt)

Parce qu'il explore couche par couche : le premier moment où il atteint un sommet correspond au plus petit nombre d'arêtes.

→ L'ordre d'exploration est donc l'ordre des distances.

3. Pourquoi cette faute est-elle difficile à repérer ? (1 pt)

Parce que le programme ne signale rien : il renvoie un chemin, simplement pas le plus court.

→ Aucune erreur n'est levée, et le résultat paraît donc correct.

4. Dans quel cas la garantie du BFS ne vaut-elle plus ? (1 pt)

Dès que les arêtes portent des poids différents.

→ Le chemin le plus court en nombre d'arêtes n'est alors plus forcément le moins coûteux.

Exercice 4 — Dijkstra

4 pts 13 min

Analyser Dijkstra

Réponds à chaque question.

LA RÈGLE DU COURS

Un algorithme prend le relais quand les arêtes portent des poids, et sa condition d'emploi n'est pas une formalité.

1. Dans quel type de graphe Dijkstra calcule-t-il le plus court chemin ? (1 pt)

Dans un graphe pondéré à poids positifs.

→ C'est le cas que le BFS ne sait pas traiter.

2. En quoi est-il une généralisation du BFS ? (1 pt)

Si toutes les arêtes valent 1, les deux algorithmes traitent les sommets dans le même ordre.

→ Le BFS est donc le cas particulier des poids égaux.

3. Que fait Dijkstra dès qu'il traite un sommet ? (1 pt)

Il fige sa distance, en supposant qu'aucun chemin ultérieur ne fera mieux.

→ C'est cette hypothèse qui rend l'algorithme efficace.

4. Pourquoi un poids négatif met-il cette hypothèse en défaut ? (1 pt)

Parce qu'un détour peut réduire la distance après coup.

→ L'algorithme renvoie alors un résultat faux sans le signaler.

Repérer une erreur et la corriger

Chaque affirmation est fausse. Écris ce qu'il fallait dire.

LA RÈGLE DU COURS

Trois fautes reviennent. Une représentation mal choisie, un parcours détourné de son usage, et une précaution oubliée.

1. « Mon graphe a un million de sommets et trois arêtes par sommet : j'utilise une matrice d'adjacence. » (1 pt)

Ce graphe est creux : la liste d'adjacence convient.→ La matrice occuperait un espace en n^2 pour stocker presque uniquement des zéros.

2. « J'ai trouvé le plus court chemin avec un parcours en profondeur. » (1 pt)

Le DFS trouve un chemin, sans garantie sur sa longueur ; seul le BFS garantit le plus petit nombre d'arêtes.

→ Le programme ne signale rien : il renvoie un chemin, simplement pas le plus court.

3. « Mon parcours de graphe tourne indéfiniment, le graphe doit être trop grand. » (1 pt)

Il manque le marquage des sommets visités : un cycle provoque une boucle infinie.

→ Il faut consulter l'ensemble des visités avant tout empilement.

Barème — ce que le correcteur regarde

Les lignes chiffrées couvrent les 5 exercices et totalisent 20 points.

EXERCICE	ATTENDU	POINTS	POINT PERDU SI
Ex.1	Représenter un graphe	5 points — 1 par question	Une matrice choisie pour un graphe creux.
Ex.2	Les deux parcours	4 points — 1 par question	Le marquage des sommets visités oublié.
Ex.3	La garantie du BFS	4 points — 1 par question	Un DFS employé pour un plus court chemin.
Ex.4	Dijkstra	4 points — 1 par question	La condition de positivité tenue pour une formalité.
Ex.5	L'erreur repérée, et la réponse juste donnée	3 points — 1 par ligne	Une correction qui garde la mauvaise structure.

ET MAINTENANT ?

16 à 20

Le chapitre est acquis. Entraîne-toi maintenant à programmer un parcours en cinq temps : la représentation selon la densité, l'objectif, la structure qui en découle, l'ensemble des visités, et la mémorisation des prédécesseurs.

10 à 15

Refais l'exercice 3 en te demandant chaque fois si le graphe est pondéré : la réponse change l'algorithme.

moins de 10

Reprends la fiche de cours et retiens ce qui sépare les deux parcours : une file plutôt qu'une pile, et rien d'autre.