



Concours national d'informatique
Épreuve écrite d'algorithmique

QUÊTE ACCEPTÉE : I BELIEVE I CAN FLY !

1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- Les exercices ne sont pas en ordre croissant de difficulté ! Lisez **vraiment** bien tout le sujet.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien, ou préparez votre présentation pour l'entretien.
- N'oubliez pas de passer une bonne journée.

Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe, sinon ça va barder.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.
- Les questions bonus sont à traiter seulement une fois que le sujet entier a été essayé.

2 Sujet

World of Prolocraft est le MEUPROG (Meilleure Excuse Utilisée en PROGrammation) le plus emblématique de l'histoire ! Bien entendu, vous vous êtes précipité dans votre magasin Macromaniac préféré pour vous emparer de ce jeu révolutionnaire et vous lancer dans l'aventure...

Partie 1 : Les stations

A peine votre personnage terminé, voilà que vous apparaissez dans le jeu. Vous êtes au beau milieu d'une cité portuaire. Un goblin vous apostrophe alors :

“ Eh ! Vous là ! Vous avez l'air d'avoir plus de neurones actifs que tous les incapables qui m'entourent réunis ! Je suis l'ingénieur en chef Vizisanie. Bienvenue à Gadgetzeurg ! J'ai un problème, et vous allez m'aider !¹ On utilisait un réseau de transports à base de griffons dans la région, pour que les aventuriers tels que vous puissent se déplacer d'une ville à l'autre facilement. On m'a chargé de les remplacer par nos hélicoptères modernes, bien plus fiables !²

J'ai déjà placé mes fidèles employés aux points stratégiques, avec nos machines les plus modernes. Mais ils sont incapables de décider où envoyer les aventuriers ! Les griffons étaient bien dressés, mais là on ne sait pas vers quelle station les envoyer. J'ai besoin de votre aide !”

Vous vous empressez d'accepter votre nouvelle mission, appâté par une récompense alléchante.

Partie 2 : Quêtes de bas niveau

Vous suivez le goblin dans son atelier³, il vous montre une carte de la région. Les emplacements des stations de vol sont marquées dessus, chacune reliée à d'autres stations, formant un réseau de transport. Vous êtes maintenant chargé de déterminer le chemin à utiliser pour déplacer les aventuriers d'une station à une autre.



FIGURE 1 – Le réseau des stations de vol de la région

Pour représenter les stations de vol, nous allons utiliser un graphe non orienté (les vols

1. Vous regardez les récompenses proposées. Vous allez l'aider.
2. Quand ils n'explorent pas...
3. Mauvaise idée

se font dans les deux sens) et pondéré (le lien entre deux stations contient une information supplémentaire : la distance entre les stations).

Le gobelin commence par vérifier vos connaissances avec quelques quêtes d'introduction, pour s'assurer que vous êtes en mesure de faire face aux défis proposés.

Question 1 (1 point)

On considère une structure de liste chaînée. Une liste peut être une liste vide, ou des données associées au reste de la liste, qui est lui-même une liste.

Écrire une fonction dans le langage de votre choix, qui permet de déterminer la taille d'une liste.

Question 2 (2 points)

Écrire une fonction qui permet de trouver le plus petit élément de la liste.

Question 3 (1 point)

On va maintenant utiliser ces listes chaînées pour représenter un graphe : chaque station est représentée par une liste contenant des références vers les stations qui lui sont adjacentes. Dans notre cas, les éléments de la liste contiendront également les distances. Vous n'êtes pas contraints d'utiliser cette représentation si vous préférez une alternative, mais vous devez alors l'expliquer.

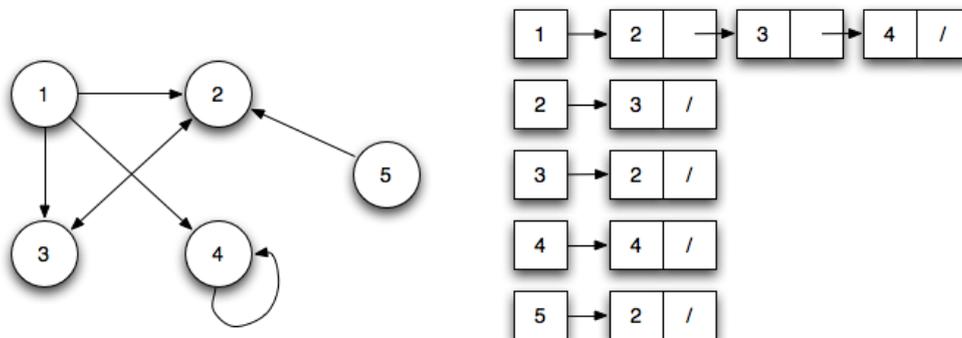


FIGURE 2 – Exemple de représentation de graphe en utilisant les listes

Écrire un algorithme qui compte le nombre de connexions totales dans le graphe.

Partie 2 : Suis moi, je connais le chemin !

Question 4 (2 points)

On veut parcourir le graphe en visitant d'abord les stations les plus proches de la station de départ, en terme de nombre de connexions. On ignore les distances sur les arêtes pour l'instant. Pour ce faire, on doit mettre de côté les stations pouvant être visitées, et les parcourir en commençant par les stations qu'on a vues le plus tôt.

Quelle structure de données peut-on utiliser pour conserver les stations pouvant être visitées ?

Question 5 (2 points)

Comment éviter de tourner en rond sur des stations déjà visitées ?

Question 6 (4 points)

Cet algorithme est appelé parcours largeur d'un graphe.

Implémentez cet algorithme

Question 7 (3 points)

On cherche maintenant à modifier l'algorithme précédent pour visiter en priorité les stations les plus proches, en prenant en compte les distances entre chaque station.

Quel changement doit être fait dans l'algorithme pour pouvoir faire ça ?

Question 8 (3 points)

Quelle nouvelle structure de données utiliser pour conserver les stations pouvant être visitées ?

Question 9 (1 point)

À quel moment peut-on arrêter de chercher en étant sûr d'avoir le résultat optimal ?

Question 10 (5 points)

Cette modification du parcours largeur est un nouvel algorithme, appelé algorithme de Dijkstra.

Implémentez cet algorithme

Question 11 (2 points)

Plutôt que d'utiliser des stations de vol comme nos points de départ et d'arrivée, on souhaite partir de coordonnées libres. Comment prendre cela en compte dans une recherche de plus court chemin ?

Question 12 (2 points)

On souhaite pouvoir utiliser des sorts de téléportation, qui permettent de rejoindre une station instantanément. Comment modifier l'algorithme pour prendre cela en compte ?

Partie 3 : Oh, les jolis spanners !

"Vous savez quoi ? On va tout reprendre ! Depuis le début ! Ce réseau a manifestement été conçu par un incapable⁴... Bon, on va quand même devoir conserver les stations, ça coûte cher à produire ces choses-là ! Vous avez juste à les connecter ensemble... Facile non ?"



FIGURE 3 – Un ensemble de stations à relier

On ne peut pas lier toutes les stations entre elles, pour des raisons de trafic aérien complexes⁶. Nos clients sont exigeants, ils demandent que le chemin parcouru ne soit jamais supérieur à t fois la distance à vol d'oiseau⁷.

4. Probablement un aventurier débutant⁵.

5. Pas vous hein ! Vous vous êtes beau. Et fort. Et merveilleux. On vous aime !

6. C'est vraiment très complexe, seuls les plus grands mages sont en mesure d'esquisser la compréhension du problème.

7. Le paramètre t , appelé élasticité, permet de régler à quel point on veut interconnecter nos stations, c'est un entier compris entre 1 et l'infini. Par exemple avec une élasticité de 1, toutes nos stations sont directement connectées, car il doit y avoir un chemin de la même longueur que la distance entre les deux stations. Avec 2, le chemin peut être deux fois plus long que la distance.



FIGURE 4 – Un mauvais réseau de stations

On propose un algorithme simple (glouton) permettant de relier les stations : pour chaque paire de stations, on les relie si la condition d'élasticité n'est pas respectée (il n'existe aucun chemin entre les deux stations de longueur inférieure à t fois la distance entre les stations). On suppose qu'on ait une fonction permettant de donner la distance du chemin le plus court entre deux stations⁸.

Question 13

(2 points)

Combien de fois appelle-t-on la fonction de distance en fonction du nombre n de stations ?

Question 14

(3 points)

On suppose que la complexité⁹ de la recherche de chemin en fonction de n est de $n \log(n)$

Quelle est la complexité en fonction de n de l'algorithme proposé ?

Question 15

(4 points)

Implémentez l'algorithme précédent.

8. Par exemple, l'algorithme de Dijkstra évoqué plus tôt dans ce sujet

9. Nombre d'opérations effectuées par l'algorithme

Partie 4 : Bonus

Question bonus 1 (7 points)

On admet qu'on a une fonction prenant en paramètre deux groupes de points, et indique s'ils sont "bien séparés". Si deux groupes de points sont bien séparés et il existe une connexion d'un groupe à l'autre, alors la condition d'élasticité est respectée sur l'ensemble du graphe si elle l'est sur chaque groupe séparément.

Proposer un algorithme plus efficace que le glouton pour relier les points. N'hésitez pas à présenter un début d'algorithme, ou à ne pas utiliser la fonction évoquée.

Question bonus 2 (2 points)

Comment prononcez-vous "Dijkstra" ?

Question bonus 3 (5 points)

Mrlgrlg glrglmglr, mrlgrlg. Rglrgmrg (mrglr glr). Gmrlgrmlg mrglmrgm ? (500 mots)

Question bonus 4 (5 points)

Quel est le meilleur archétype de personnage dans un jeu de rôle ? Illustrer avec des schémas.

Le sujet comporte 6 pages (sans compter la page de garde) et 15 questions, parmi lesquelles 4 questions bonus. Les questions normales sont notées sur 37 points, et les questions bonus rapportent au total 4 points, plus 1 point de présentation.