



Concours national d'informatique

Épreuve écrite d'algorithmique

PROLOGIN CONTRE-ATTAQUE

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.
- 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

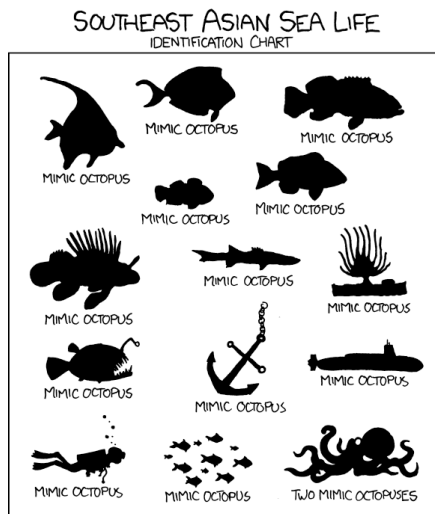
- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez 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.

2 Une (sombre) histoire de décapodes

Ces dernières années, Prologin a un gros problème : à chaque finale, les ballons de baudruche disparaissent. On ne sait pas vraiment à quoi ces ballons servent, mais une chose est sûre : ils sont d'une importance capitale.

Cette année, les organisateurs ont décidé d'agir ! Ils veulent construire la décoration de la finale de façon à ce qu'il soit impossible au voleur¹ de ballons de s'échapper.

Pour des raisons de confidentialité, nous appellerons maintenant le malfaiteur : calamar^{2 3}.



3 Pêche du calamar⁵

Dans un premier temps, par manque de moyens humains, l'association Prologin fait appel à Mme. Lamia⁶, une détective privée qui peut donner à coup sûr le numéro de l'ordinateur sur lequel se trouve le calamar. Elle ne peut cependant pas vous dire le nom du calamar.

Pour aider Mme. Lamia dans sa tâche, vous disposez de la liste des candidats L contenant des tuples (numero, nom_candidat).

Question 1 (1 point)

Écrivez une fonction `nom_calamar(L, numero)` qui, pour une liste de candidats L et un `numero` donné retourne le nom du calamar.

Question 2 (1 point)

Donnez la complexité (en fonction de la taille de L) de votre algorithme pour la Question 1, c'est-à-dire le nombre d'opérations effectuées par l'algorithme.

Question 3 (3 points)

On considère maintenant que la liste des candidats est triée par numéro. Réécrivez une version plus efficace de l'algorithme de la Question 1.

Question 4 (1 point)

Donnez la complexité (en fonction de la taille de L) de votre algorithme pour la Question 3.

1. Nous savons de source proche de l'enquête qu'il s'agit des méfaits d'une seule personne
2. vert
3. Pour aucune raison particulière⁴
4. cacahuète
5. L'association Prologin soutient seulement la pêche de calamar métaphorique
6. fille du sénateur éponyme

4 Enter the Matrice⁷

Pour simuler ce vol de ballons, les organisateurs décident de représenter cette situation comme un graphe où les sommets sont les positions possibles de l'organisateur et du calamar, et les arêtes les changements de position possible pour un sommet donné.

L'organisateur et le calamar se déplacent sur le graphe chacun leur tour en commençant par l'organisateur.

Malheureusement, l'organisateur choisit d'abord sa position initiale dans le graphe, puis le calamar choisit à son tour sa position sur le graphe, ce qui lui donne un avantage certain.

Définition 1. *Un graphe est une structure de données permettant de représenter des relations dans un ensemble d'éléments. Un graphe est composé de sommets, qui représentent les éléments, et d'arêtes, qui représentent les relations entre les différents éléments.*

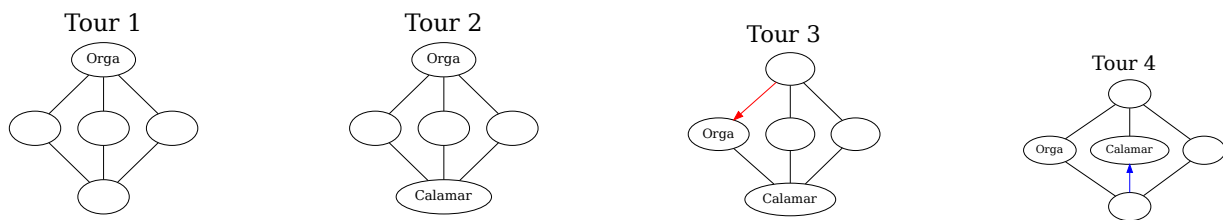


FIGURE 1 – Un exemple de mouvement de l'organisateur et du calamar

Question 5

(2 points)

Quelle structure de données peut-on utiliser pour représenter votre graphe (plusieurs réponses possibles) ?

Pour l'instant, considérons que la décoration de la finale consiste seulement en un gonflable, et que l'organisateur et le calamar peuvent uniquement tourner autour de ce dernier.

Définition 2. *Un cycle dans un graphe est un chemin en suivant les arêtes où l'on revient à son sommet de départ, sans passer plusieurs fois par le même sommet. La longueur d'un cycle est le nombre de sommets parcourus en suivant ce cycle.*

Question 6

(3 points)

Quelle est la longueur maximale du cycle pour que l'organisateur attrape à coup sûr le calamar et pourquoi ?

Maintenant que les organisateurs commencent à comprendre ces histoires de graphe, ils décident d'utiliser des décorations plus complexes. Le graphe peut donc prendre n'importe quelle forme.

Question 7

(9 points)

Écrivez une fonction `calamar_libre(G, N)` qui, pour un graphe `G` donné, retourne `True` si le calamar peut s'échapper. Si le calamar court toujours au bout de `N` tours on considérera que le calamar peut s'échapper.

7. Entrer dans la Matrice

Question 8

(1 point)

Donnez la complexité de votre fonction `calamar_libre(G, N)`.

5 Le calamar⁸ sur l'écueil⁹

On définit un écueil comme un sommet où le calamar ne peut pas s'échapper au prochain tour si l'organisateur est sur un sommet adjacent.

Définition 3. *Deux sommets sont adjacents si et seulement si ils partagent une arête commune.*

Question 9

(1 point)

Écrivez une fonction `ecueil(G)` qui, pour un graphe `G` donné retourne la liste des écueils du graphe.

Question 10

(1 point)

Donnez la complexité de votre fonction `ecueil(G)`.

Après de longues recherches, les organisateurs de Prologin se rendent compte qu'il est possible de savoir si un graphe permet au calamar de s'échapper ou si l'organisateur peut attraper le calamar au bout d'un certain temps.

Il suffit juste de supprimer tous les écueils jusqu'à se retrouver avec un graphe sans aucun écueil. Si ce graphe n'a qu'un sommet alors le calamar se fera attraper, si ce n'est pas le cas le calamar pourra toujours s'échapper.

Question 11

(1 point)

Réécrivez une version plus efficace de la fonction `calamar_libre(G)` (sans limite de tours cette fois-ci).

Question 12

(1 point)

Donner la complexité de votre fonction `calamar_libre(G)`.

6 Prologin contre-attaque

Après avoir étudié de nombreuses configurations de décoration, les organisateurs de Prologin se rendent compte que dans la plupart des cas, le calamar peut s'échapper. Comme ces histoires de vols de ballons sont de la plus haute importance pour le Président de l'association, il veut à tout prix attraper le calamar. Mais comme la plupart des organisateurs sont très occupés à "travailler" dans le gonflable, il voudrait savoir quel est le nombre d'organisateur à déployer sur le graphe pour être sûr d'attraper le calamar.

Durant leurs recherches, les organisateurs ont aussi découvert que pour tout graphe planaire, le nombre d'organisateur minimum pour attraper le calamar à coup sûr est de 3.

Définition 4. *Un graphe est dit planaire s'il peut être représenté sur un plan sans qu'aucune arête n'en croise une autre.*

Question 13

(1 point)

Écrivez une fonction `est_planaire(G)` qui, pour un graphe `G` donné, retourne `True` si le graphe est planaire.

De plus, si le plus petit cycle d'un graphe a pour longueur 5 ou plus, alors le nombre d'organisateur minimum pour attraper le calamar est égal au degré minimal du graphe.

Définition 5. *Le degré minimal d'un graphe est le nombre minimal d'arêtes connectées à un sommet.*

9. Mollusque céphalopode comestible, de forme allongée, caractérisé par sa coquille dorsale réduite à une plume cornée. (Il est voisin de très grandes formes abyssales connues seulement par les blessures qu'elles infligent aux cachalots.)

9. Tête de roche couverte par moins de 20 m d'eau, parfois à fleur d'eau, et constituant un danger pour la navigation. (Sous les hautes latitudes, les côtes taillées dans les roches dures sont parfois précédées de jardins d'écueils.)

Question 14

(3 points)

Écrivez une fonction `longueur_min(G)` qui, pour un graphe G donné, retourne la longueur du plus petit cycle. Donnez la complexité de la fonction.

Question 15

(2 points)

Écrivez une fonction `degre_minimal(G)` qui, pour un graphe G donné, retourne le degré minimal du graphe, puis donnez sa complexité.

Question 16

(1 point)

Écrivez une fonction `borne_orga(G)` qui, pour un graphe G donné, retourne une borne inférieure et supérieure pour le nombre minimum d'organiseurs pour attraper le calamar.

7 Section Bonus

Cette section est facultative et comporte des questions plus difficiles, il est déconseillé de commencer à y répondre avant d'avoir bien fini les questions qui précèdent.

Question bonus 17

(1 point)

Dessinez une pieuvre mimétique.

Question bonus 18

(3 points)

Prouver qu'enlever les écueils d'un graphe permet de déterminer s'il permet au calamar de s'échapper.

Question bonus 19

(3 points)

Prouver que si un graphe est planaire, alors le nombre d'organiseurs maximum pour attraper le calamar est de 3.

Question bonus 20

(42 points)

Pour un graphe à n sommets, quel est le nombre d'organiseurs minimum qu'il faut déployer pour attraper le calamar (en fonction de n) ?

Le sujet comporte 5 pages (sans compter la page de garde), 20 questions, et 49 questions bonus. Les questions normales sont notées sur 32 points, et les questions bonus rapportent au total 49 points, plus 1 point de présentation.