



Concours national d'informatique
Épreuve écrite d'algorithmique
Paris I

Dimanche 31 janvier 2016

QUE LA FORCE SOIT AVEC TOI



**COME TO THE
DARK SIDE
(WE HAVE COOKIES!)**

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.
- 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.
- 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 Sujet

Introduction

Il y a bien longtemps, dans une galaxie lointaine, très lointaine...

Une bataille faisait rage depuis des milliers d'années, brûlant monde après monde, peuple après peuple, et dont l'inévitable avancée menaçait toute la galaxie.

Au cœur de la planète Coruscant, dernier bastion de la résistance, le conseil Jedi se réunissait, peut-être pour la dernière fois. Les tranchées stellaires cédaient une à une. Le temps était compté. Il ne s'agissait que d'une question d'heures avant que l'armée de Sith ne parvienne à pénétrer la cité et attaque le temple.

Le chef de l'assemblée Jedi entra dans la salle, brandissant ce qu'il savait être leur dernier espoir¹, le fruit d'années d'espionnage de l'Empire. Le relevé de midichlorolémie² de chaque Sith.

L'art du combat Jedi, c'est bien connu, se résume principalement à avoir un taux de midichloriens plus élevé que son adversaire³. Le conseil a besoin de votre aide pour établir une stratégie de combat à partir de ces informations cruciales.

Partie 1

« Personne, par la guerre, ne devient grand. »

— *Star Wars, épisode V : L'empire contre-attaque*

Lors d'un affrontement entre un Jedi et un Sith, c'est toujours le combattant qui a la plus grande force⁴ qui remporte le combat⁵.

L'équilibre dans la force ayant été rétabli, on sait qu'il y a toujours autant de Sith que de Jedi dans un combat. Le conseil décide donc d'assigner à chaque Jedi un unique Sith à combattre, afin de maximiser le nombre de duels remportés, et par conséquent minimiser les pertes. Des vies sont en *jeu*⁶ !

Dans cette partie, on dispose d'un tableau `Jedi` et d'un tableau `Sith` tels que `Jedi[i]` correspond à la force du i -ème Jedi et `Sith[i]` à celle du i -ème Sith.

On suppose, dans la partie 1, qu'aucun combat ne se solde par une égalité. Ainsi, jamais deux forces ne sont égales.

Question 1

(1 point)

Écrire une fonction `compare` qui prend en argument deux entiers i et j désignant des combattants et qui renvoie `vrai` si le Jedi i gagne le combat contre le Sith j , `faux` sinon.

1. *Non... il y en a un autre.*

2. Ou le taux de midichloriens, si vous préférez.

3. Si vous vous arrachez déjà les cheveux à la lecture de cette phrase, attendez un peu la suite.

4. Oui oui, la midichlorolémie.

5. *La raison du plus fort est toujours la meilleure, car tout se réduit à des questions de force.*

6. Vous l'attendiez, hein ?

Question 2

(2 points)

Écrire une fonction `nombre_de_duels_gagnes` qui renvoie le nombre de duels gagnés par les Jedi si chaque Jedi i combat le Sith i .

Pour établir une stratégie gagnante, le conseil cherche à attribuer des adversaires à chaque Jedi pour obtenir le meilleur résultat possible.

Question 3

(2 points)

Écrire une fonction `attribue_un_adversaire` qui prend en argument i et renvoie le numéro j d'un Jedi qui pourrait vaincre le Sith i , sauf si ce Sith est plus fort que tous les Jedi⁷.

Question 4

(4 points)

En vous aidant ou vous inspirant des questions précédentes, écrire une fonction `attribue_chaque_adversaire` qui renvoie un tableau \mathbf{t} tel que $\mathbf{t}[i]$ est le numéro du Sith que devra combattre le Jedi i afin qu'il y ait un maximum de combats gagnés. Justifier.

Partie 2

« Et c'est ainsi que s'éteint la liberté, sous une pluie d'applaudissements. »

— *Star Wars, épisode III : La Revanche des Sith*

La bataille des tranchées stellaires venait de prendre fin. Les deux camps avaient subi de nombreuses pertes, mais l'armée Sith avait néanmoins gagné du terrain. Le temple, ainsi que le reste de la république, étaient toujours menacés.

Il n'y a à présent plus le même nombre de Jedi que de Sith⁸. On suppose maintenant que la force n'est plus quantifiable⁹. En effet, après ces affrontements, les belligérants ont commencé à améliorer leurs techniques d'attaque et à repérer les points faibles de leurs adversaires.

Ainsi, on connaît maintenant uniquement l'issue des combats entre deux individus. Un combat peut être soit gagné, soit perdu, soit à issue incertaine (autrement dit, une égalité).

Question 5

(2 points)

Quelle structure peut-on utiliser à présent afin de représenter ces données ?

Le conseil Jedi aimerait malgré tout avoir un ordre d'idée de la puissance de chaque combattant afin de les classer, c'est à dire de pouvoir les trier de la façon suivante :

- Si un Jedi i bat un Sith j , il apparaîtra avant lui dans la liste.
- Si un Sith i bat un Jedi j , il apparaîtra avant lui dans la liste.
- Si un combat est indéterminé, n'importe lequel des deux combattants peut apparaître en premier.

7. Et là, on va avoir un problème...

8. Bien évidemment, les proportions obtenues dépendent de la validité de vos algorithmes de la partie I.

9. C'est probablement mieux ainsi, vous en conviendrez.

Question 6

(1 point)

Est-il toujours possible d'obtenir une telle liste ?

Si non, justifiez par un exemple. Dans la suite, vous pourrez supposer que nous n'avons pas affaire à ce cas de figure¹⁰.

Question 7

(3 points)

Montrer que si l'on a un ordre sur les Jedi et les Sith, on peut rajouter dans n'importe quel ordre au début de la liste un certain nombre de Jedi et de Sith qui ne sont battus par personne (tous les combats qui les impliquent se résolvent donc soit par une victoire pour eux soit par une égalité) tout en gardant les propriétés du tri valables.

Question 8

(5 points)

Écrire une fonction `tri_force`¹¹ qui donne une liste répondant au problème du conseil Jedi.

Question 9

(3 points)

Comment modifierez-vous l'algorithme pour détecter les cas où, comme dans la question 7, il n'est pas possible de construire une telle liste ?

Partie 3

« J'ai l'impression que mes yeux vont mieux. Au lieu d'un grand flou noir, je vois un grand flou lumineux. »

— *Star Wars, épisode VI : Le Retour du Jedi*

On suppose dans cette partie que l'issue de tous les combats peut être déterminée : il ne peut de nouveau plus y avoir d'égalité. Cependant, il n'y a pas forcément d'ordre transitif dans les victoires, c'est à dire qu'une chaîne de combattants reliés par des victoires consécutives n'aboutit pas forcément à une victoire entre le premier et le dernier. Par exemple, si Luke bat Dark Vador qui bat Obi wan Kenobi, qui bat le Général Grievous¹² ce dernier pourrait tout de même battre Luke¹³.

Afin d'y voir plus clair, le conseil Jedi se penche sur le cas que nous venons d'énoncer qu'il représente sous la forme du schéma de la page suivante, où une flèche représente une victoire et une absence de flèche représente une défaite.

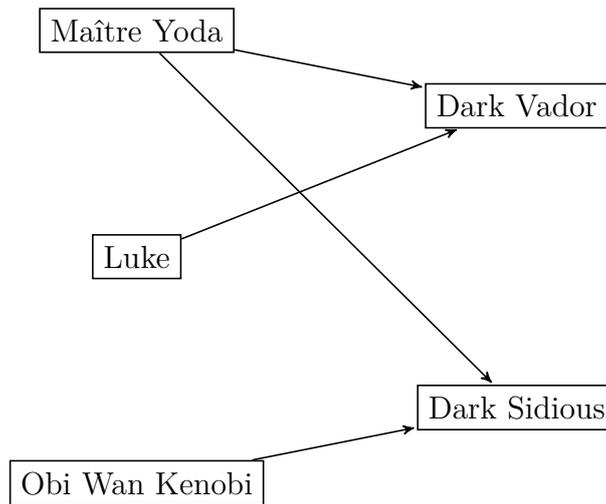
10. Si vous pensez qu'il n'existe pas, cette supposition devient évidente.

11. \triangle

$\triangle \triangle$

12. Certes, il ne sait pas utiliser la force, mais il a des sabres laser. Oui, c'est absurde.

13. S'il n'était pas mort après la course poursuite inutile, anodine, stérile, terne, nulle, fatigante, sans conséquence et visuellement épuisante du milieu de l'épisode III.



On dispose d'une structure lpdj ¹⁴ tel que le i -ème élément vaut j si le Jedi i doit combattre le Sith j . On se donne également la structure inverse lpcs ¹⁵.

Sur le schéma ci-dessus, on remarque qu'il peut y avoir des chemins (sans tenir compte du sens des flèches, qui sont toujours dans le sens des Jedi vers les Sith) alternant entre Jedi et Sith, et passant au plus une fois par personnage, par exemple :

Obi Wan Kenobi \rightarrow Dark Sidious \rightarrow Maître Yoda \rightarrow Dark Vador

Si on avait décidé que Yoda devrait combattre Dark Sidious, on peut améliorer le tout en réassignant Vador à Yoda et Sidious à Obi Wan.

Question 10

(6 points)

En vous inspirant de cet exemple, donner en justifiant un algorithme permettant de trouver un chemin tel que :

- Le premier point est un Jedi qui n'est assigné à aucun combat ;
- Le dernier point est un Sith qui n'est assigné à aucun combat ;
- Tous les personnages intermédiaires sont assignés ;
- Exactement une arête sur deux correspond à une assignation.

Écrire une fonction **inverse** qui, à partir d'un tel chemin, modifie lpdj et lpcs de manière à avoir une victoire supplémentaire pour les Jedi.

Question 11

(5 points)

En déduire une fonction **assigner** qui modifie lpdj et lpcs , pour maximiser le nombre de duels gagnés par les Jedi.

Question 12

(10 points)

On remarque qu'il est possible d'accélérer l'algorithme de la question précédente en cherchant plusieurs chemins en même temps. Écrire cette nouvelle fonction **chemins** et adapter l'algorithme de la question précédente pour tenir compte de cette modification.

14. Le plan des Jedi.

15. Le plan contre les Sith.

Question 13

(8 points)

Quelle est la complexité de l'algorithme de la question 13 ? Combien de temps prendrait votre algorithme à s'exécuter sur un sabre laser cadencé à 1337 Ghz alimenté par une pile de 12 Gigawatts, pour une liste initiale de 11 milliards de Jedi ? Justifier.

Partie bonus

Ne traitez les questions bonus que si tout le reste du sujet terminé vous avez.

Question bonus 14

(20 points)

Vous disposez maintenant de données sur les combats entre Jedi, plus précisément, de la liste des combats à issue certaine entre deux Jedi. Cependant, vous ne connaissez pas l'issue de ces affrontements. Décrire un algorithme fournissant pour chaque combat déterminé de cette liste un résultat (victoire ou défaite) plausible, c'est-à-dire de sorte que :

- un tri des Jedi au sens de la partie 2 soit possible ;
- si un Jedi en bat un autre, qui lui-même en bat un troisième, alors le premier bat le troisième de façon certaine.

Question bonus 15

(1 point)

À quels personnages doit-on attribuer les répliques au début de chaque partie du sujet ?

Question bonus 16

(1 point)

“So then Georges Lucas completely and utterly finally ruins Star Wars forever by having Qui-Gon explain that the force is microscopic organisms.”

— *The Phantom Menace Review*, Harry S. Plinkett

Pensez-vous que les midichloriens sont un ajout utile au canon de Star Wars ? Argumentez en vous appuyant sur les questions précédentes. Vous développerez une argumentation en 3 parties traitant du sujet en 500 mots (± 50).

Written and Directed by
PROLOGIN¹⁶

Le sujet comporte 6 pages (sans compter la page de garde) et 16 questions, parmi lesquelles 3 questions bonus. Les questions normales sont notées sur 52 points, et les questions bonus rapportent au total 22 points, plus 1 point de présentation.

16. Tum tudum tudum tu dudududu duuuuuuum duuuuuuum dududududuuuuuuum dum...