



# Concours National d'Informatique

Sujet de demi-finale Paris 2 / Lille

21 février 2009

# Métagrammes

## 1 Préambule

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

Ceci est une épreuve d'algorithmique. Cela signifie que ce qui intéresse les correcteurs n'est pas la manière dont vous écrivez votre code, qui sera testée cet après-midi, mais votre manière de réfléchir et de résoudre des problèmes.

À ce titre, tous les langages sont autorisés, y compris le pseudo-code, pourvu que vous indiquiez lequel vous utilisez. Si vous éprouvez une quelconque difficulté avec votre langage, vous pouvez ainsi expliquer « en français » votre manière de résoudre la question, à condition que vous donniez un processus détaillé facilement transposable en un programme.

### 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 la demi-finale et la date, en haut de votre copie.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le nous, nous pouvons ajouter des questions plus difficiles. Mais vous le regretterez peut-être.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- 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.
- Les questions de ce sujet ont été triées dans un ordre de difficulté aléatoire. Ne restez donc pas bloqué sur un exercice, vous réussirez peut-être à faire la suite.

## 2 Sujet

### Introduction

Aujourd'hui, Sophie s'ennuie. C'est un jour férié, le premier mai, et elle a fini tous ses devoirs depuis un moment. Elle voudrait bien sortir, aller au cinéma, jouer au bowling, boire du Champomy dans un bar, mais avec qui ? Son amie Cimone ne lui envoie plus de SMS depuis un mois, et ses autres copines ne sont pas disponibles. Elle voudrait bien regarder une série idiote à la télévision, mais son poste est tombé en panne ce matin. En regardant son calendrier, elle remarque que l'on est un vendredi treize ; comme quoi, ce jour porte réellement malheur. Dans la matinée, elle a déjà rangé sa chambre, trié ses chaussettes et fait toutes ces choses que l'on reporte habituellement au lendemain.

Aujourd'hui, Sophie s'ennuie. Alors, elle a cherché un jeu. Plongeant profondément dans ses souvenirs, elle se rappelle vaguement les règles d'un jeu de lettres, où il faut passer d'un mot à un autre, en changeant une lettre à la fois. Elle se dit qu'il manque quelque chose, mais elle souhaite tester dès maintenant.

### Question 1 (2 points)

On vous donne un mot A et un mot B de même longueur. En changeant une lettre de A à chaque tour, on souhaite arriver au mot B. Il y a bien sûr une infinité de possibilités. Écrivez une fonction qui renvoie le nombre de tours nécessaires pour la solution la plus courte.

Par exemple, pour passer de *chat* à *cran*, les solutions les plus courtes sont : *chat* – *crat* – *cran* et *chat* – *chan* – *cran*. La fonction doit donc renvoyer 2 pour ce cas.

### Question 2 (1 point)

Sophie se rend compte que ce n'est pas très amusant. Heureusement pour elle (et pour vous), sa mémoire lui revient : il faut que chaque mot intermédiaire soit un mot valide du dictionnaire. Les verbes peuvent être conjugués, les adjectifs accordés, et ainsi de suite mais, pour simplifier un peu, on ignore les accents.

Par exemple, pour passer du coq à l'âne, une bonne solution est : *coq* – *col* – *cil* – *ail* – *aie* – *âne*. Ce jeu, inventé par Lewis Carroll, s'appelle les Doublets. En linguistique, on appelle ce genre de transformation un métaplasme ou métagramme. Dans la suite du sujet, on dira simplement que *col* est un voisin de *coq*.

Pour montrer que vous avez compris le jeu, passez de *homme* à *femme*<sup>1</sup>.

---

<sup>1</sup>Vous pouvez écrire n'importe quelle solution, mais sachez que la plus courte tient en trois transformations.

### Question 3 (1 point)

Sophie vient de trouver une liste des mots du français (hélas, ils ne sont pas triés, mais leurs accents ont été retirés). Proposez une structure de données pour représenter cette collection de mots.

Vous supposerez par la suite que les fonctions permettant de manipuler cette collection sont fournies.

### Question 4 (2 points)

Sophie souhaite savoir quels mots peuvent être joués, juste après le mot donné. À partir de ce mot et d'une collection de mots, écrivez une fonction qui renvoie l'ensemble des mots autorisés.

Par exemple, avec le mot *singe*, votre fonction devra renvoyer *songe*, *siège* et *linge*.

### Question 5 (1 point)

Sophie est ravie ; elle choisit deux mots au hasard et, grâce à votre fonction, elle cherche des solutions du jeu. Elle en est tellement contente qu'elle voudrait montrer cela à son cousin de Laponie, qu'elle verra la semaine prochaine. Mais, se dit-elle, c'est qu'il ne parle pas français !

Vous devez aider Sophie : indiquez tout le travail qu'il y a à faire pour que le programme gère le lapon. Réécrivez toutes vos fonctions qui doivent l'être.

### Question 6 (3 points)

Après de nombreux tests, Sophie s'aperçoit que votre fonction est trop lente. Elle pense qu'en modifiant la structure de données, il devrait être possible d'améliorer sensiblement les performances, au niveau du temps d'exécution. Sophie a en effet l'intention d'appeler la fonction de la question 4 des milliers de fois avec le même dictionnaire.

Proposez une nouvelle structure de données pour le dictionnaire, dans le but d'optimiser la recherche de mots voisins. Vous supposerez par la suite que les fonctions de manipulation usuelles sont déjà écrites (insertion d'un élément...).

Justifiez votre choix en indiquant les avantages et les inconvénients de cette structure. N'oubliez pas de tenir compte à la fois de la vitesse d'exécution et de la consommation mémoire.

### Question 7 (2 points)

Écrivez la fonction permettant de convertir la structure de la question 3 en celle de la question 6. Réécrivez ensuite la fonction de la question 4 en utilisant cette nouvelle structure.

### Question 8 (5 points)

Cela fait plus d'une demi-heure que Sophie cherche à passer de *chaud* à *froid*. Elle n'y arrive pas et vous demande de l'aide. Écrivez donc une fonction qui, à partir du mot A et du mot B, renvoie la meilleure solution (si plusieurs solutions sont aussi courtes, choisissez celle que vous préférez<sup>2</sup>), sous forme d'une liste de mots.

Par exemple, avec *singe* et *signe*, en utilisant un dictionnaire français, votre fonction doit renvoyer : *singe*, *linge*, *liage*, *liane*, *ligne* et *signe*.

Si cette question vous semble trop difficile, essayez déjà de renvoyer la longueur du chemin le plus court.

### Question 9 (2 points)

Sophie teste votre fonction pour trouver la solution pour *chaud* et *tiède*. Estimez son temps d'exécution, sur l'ordinateur rose de notre héroïne, sachant que chaque mot possède environ trois voisins possibles, que la chaîne la plus courte passe par 15 mots et que le dictionnaire contient 300 000 mots. Estimez également la quantité de RAM nécessaire (grossièrement).

### Bonus

Vous avez parfaitement réussi les questions précédentes, vous vous êtes déjà relu cinq fois et il vous reste deux heures avant la fin de l'épreuve ? Pas de problème, voici de quoi vous occuper. Les questions qui suivent ne rapportent de points que si les questions précédentes ont été réussies.

### Question 10

Passez de *deux* à *sept*, en utilisant les mots français de votre connaissance.

### Question 11

Diantre ! se dit Sophie, même la fonction de la question 8 ne trouve aucun chemin entre *chaud* et *froid*. Après quelques observations, elle se rend compte que, si plus de quatre mille mots sont accessibles à partir de *chaud*, *froid* ne possède aucun voisin possible. Pourtant, on constate qu'il existe un chemin entre *chaude* et *froide*. Il suffirait donc d'autoriser l'insertion et la suppression de lettres, mais seulement quand il n'y a pas d'autre solution.

Adaptez donc votre fonction de la question 8. Vous devez renvoyer le chemin le plus court, en minimisant d'abord le nombre d'insertions.

---

<sup>2</sup>Certaines personnes préfèrent la solution passant par *nutella*, plutôt que par *épinard*, mais vous faites ce que vous voulez.

Par exemple, si on appelle votre fonction avec *poule* et *polen*, votre fonction doit renvoyer le chemin faisant six remplacements de lettres successifs, plutôt que de faire *poule* – *pole* – *polen*.

## Question 12

Passez maintenant de *sept* à *huit*.

Un point de présentation vous sera accordé si vous faites un effort pour rendre un travail clair et lisible. La fin de cette page est laissée blanche pour vous permettre de faire un joli dessin.