Université Paris 7-Denis Diderot
L2 — Eléments d'algorithmique

Examen du 25 octobre 2014.
Durée 3h — aucun document autorisé

Avertissement :
Les programmes doivent être écrits avec une syntaxe claire et précise.
Ne seront corrigés que les programmes commentés.

On dispose de cartes sur lesquelles sont dessinés trois traits reliant 3 points sur le bord gauche à 3 points sur le bord droit. De chacun des points du bord gauche part un et un seul trait, et à chacun des points du bord droit n'arrive qu'un seul trait. De plus chacun des trois traits est d'une couleur différente des autres. Les couleurs sont prises dans l'ensemble $\{jaune, rouge, vert\}$.

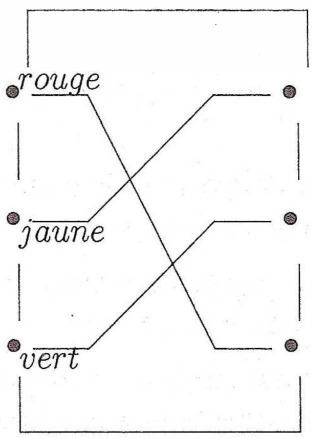

Figure 1: une carte

Question 1 : Combien y a-t-il de cartes différentes ?

Dans la suite, on propose de coder les couleurs $\{jaune, rouge, vert\}$ par les entiers 1, 2 et 3 respectivement, et de coder les cartes par un 6-uple $(x_1, x_2, x_3, x_4, x_5, x_6)$, représentant les couleurs du bord droit puis du gauche, et pour chaque côté de haut en bas. Ainsi, la carte de la figure 1 est-elle représentée par le 6-uple $(2, 1, 3, 1, 3, 2)$.

Comme les points du bord gauche sont situés aux mêmes hauteurs (globalement) que les points du bord droit, si l'on met côte à côte 2 telles cartes, les points de droite de la carte la plus à gauche coïncident avec les points de gauche de la carte à droite, mais les couleurs ne se correspondent pas forcément.

Question 2 : Proposer une fonction qui prend en argument les codages de deux cartes, et renvoie 1 si les couleurs des points de droite de la première carte correspondent exactement aux couleurs des points de gauche de la seconde, et 0 sinon.

Dans toute la suite, un entier n est fixé à l'avance (par exemple $n = 16$), et on suppose que l'on dispose d'un jeu de n cartes, toutes différentes, c'est-à-dire que l'on a la donnée de n codages de cartes dans un tableau. Chaque carte est alors

Pour tout entier k avec $k \leq n$, on peut placer côte à côte k cartes parmi les cartes du jeu. On appelle une telle disposition de cartes une *tentative* de longueur k.

Question 3 : Vérifier que l'on peut représenter une tentative par une suite d'entiers : les numéros des cartes choisies. Proposer une structure susceptible de contenir la représentation d'une tentative.

Une tentative est *correcte* lorsque chaque fois que deux cartes sont côte à côte, les couleurs se correspondent.

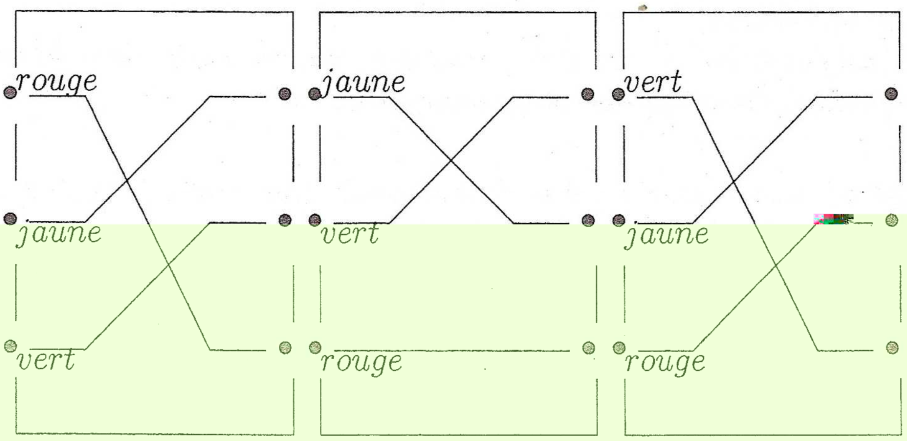

Figure 2: une tentative correcte

Question 4 : Proposer une fonction qui prend en argument la représentation d'une tentative, et renvoie 1 si elle est correcte, et 0 sinon.

Une tentative est *complète* lorsque sa longueur vaut n. Une tentative complète qui est correcte est appelée une *réussite*.

Question 5 : Définir un ordre total sur les tentatives complètes, et en déduire un algorithme qui énumère toutes les tentatives complètes. Combien y en a-t-il ?

Question 6 : Déduire directement de cette énumération, et de la fonction qui teste si une tentative est réussie, un algorithme qui résout le problème :

Problème de la réussite
Donnée : la représentation d'un jeu de n cartes
Question : peut-on faire une réussite avec ce jeu ?

Question 7 : Quelle est la complexité de cet algorithme ?

Bien entendu, si une tentative de longueur inférieure à n n'est pas correcte, il est impossible de la prolonger en une réussite.

Question 8 : Faire une énumération de toutes les tentatives correctes (quelque soient leurs longueurs). Quelle méthode du cours utilisez-vous ?

Question 9 : En modifiant le programme précédent, donner un nouvel algorithme qui résout le problème de la réussite.

Questions subsidiaires :

Question 10 : Expliciter tout ce qu'il faut modifier dans ce qui a été fait si l'on suppose maintenant que les bords des cartes ont p points sur chaque bord, et que [illegible]

[illegible] aux cartes, combien y a-t-il de tentatives complètes ? Expliciter tout ce qu'il faut modifier alors dans [illegible]

[illegible]

Question 12 : Redéfinir pour ce nouveau jeu les notions de tentative et de réussite, dénombrer les tentatives complètes, et résoudre le problème de la réussite avec la règle du jeu qui interdit de faire tourner une carte sur elle-même, puis avec la règle qui l'autorise.