

SESSION 2020

**CAPES
CONCOURS EXTERNE
ET CAFEP CORRESPONDANTS**

Section : NUMERIQUE ET SCIENCES INFORMATIQUES

SECONDE ÉPREUVE D'ADMISSIBILITÉ

Durée : 5 heures

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout matériel électronique est rigoureusement interdit.

Si vous repérez ce qui vous semble être une erreur d'énoncé, vous devez le signaler très lisiblement sur votre copie, en proposer la correction et poursuivre l'épreuve en conséquence. De même, si cela vous conduit à formuler une ou plusieurs hypothèses, vous devez la (ou les) mentionner explicitement.

NB : Conformément au principe d'anonymat, votre copie ne doit comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé consiste notamment en la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de la signer ou de l'identifier.

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie.

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

► Concours externe du CAPES de l'enseignement public :

Concours	Section/option	Epreuve	Matière
E B E	6 2 0 0 E	1 0 2	0 5 3 0

► Concours externe du CAFEP/CAPES de l'enseignement privé :

Concours	Section/option	Epreuve	Matière
E B F	6 2 0 0 E	1 0 2	0 5 3 0

Préambule : le sujet comporte 5 parties. L'ensemble du sujet porte sur une même thématique qui est la recherche d'une information satisfaisant certaines contraintes. Cette thématique est illustrée sur l'exemple du Mastermind, en parties II., III. et IV. Les notations sont communes à toutes les parties et sont introduites au fur et à mesure du sujet. Les parties peuvent être traitées de façon indépendante, sauf pour la partie III. où on pourra réutiliser ou admettre les notations et résultats de la partie II., et pour la partie IV. où on pourra réutiliser les notions, contenus, algorithmes et fonctions Python déterminés dans les parties II. et III.

Dans ce sujet, certains contenus sont à destination des élèves de lycée. Ces contenus sont en italique dans le sujet.

Pour certaines questions, une longueur de réponse attendue est indiquée. Nous vous invitons à respecter ces indications.

Partie I. Le trésor et la fausse pièce

Vous avez proposé à vos classes de première en spécialité NSI, le projet ci-dessous.

Vous êtes un ou une pirate qui vient de récupérer sa part du trésor constitué d'un beau tas de pièces d'or. Hélas, vous avez vent d'une supercherie : l'une des pièces n'est pas en or massif, mais en un alliage plus léger que l'or et est simplement plaquée or, pour qu'on ne puisse pas la distinguer visuellement des autres pièces. Le seul signe distinctif est qu'elle est plus légère que les autres. Vous n'avez pas à votre disposition un atelier de chimiste ni un laboratoire scientifique vous permettant d'étudier très précisément ces pièces, mais simplement une balance à plateaux sans aucun poids, comme celle donnée ci-dessous :



Pouvez-vous, avec cette simple balance, retrouver la fausse pièce ?

I.1. L'une de vos classes a été beaucoup plus rapide que les autres et les élèves ont très rapidement proposé une réponse utilisant une approche par dichotomie. Pour alimenter leur curiosité et leur créativité, vous voulez les amener à une solution, dite *par trichotomie*, utilisant un découpage en 3 ensembles (contrairement à 2 ensembles utilisés avec l'approche dichotomique). Afin de les aider, vous préparez un énoncé donné ci-dessous (partie en italique) et que vous comptez leur distribuer. **Rédiger un corrigé de cet énoncé, tel que vous le proposeriez à vos élèves.**

Exemples pour un petit nombre de pièces dans le trésor.

- 1. Expliquer comment procéder avec 2 pièces pour retrouver la fausse pièce.*
- 2. Expliquer comment procéder en une seule pesée quand il y a 3 pièces.*
- 3. Expliquer comment conclure en deux pesées quand il y a 4 pièces.*

Examinons maintenant comment faire avec 9 pièces. On procède en divisant les 9 pièces en 3 tas, et selon la réponse de la balance on identifie le tas contenant la pièce plus légère, comme indiqué sur la figure 1.

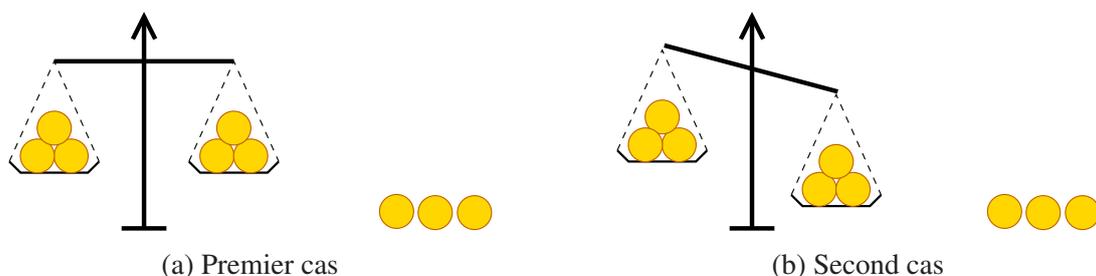


FIGURE 1 – Exemple avec 9 pièces.

Dans le cas illustré en figure 1a, le tas contenant la pièce la plus légère est le tas qui n'est pas sur un plateau. On identifie, en une pesée supplémentaire, la pièce la plus légère parmi les 3 pièces de ce tas.

Dans le cas illustré en figure 1b, le tas contenant la pièce la plus légère est le tas qui est sur le plateau de gauche. On identifie, en une pesée supplémentaire, la pièce la plus légère parmi les 3 pièces de ce tas.

On peut donc conclure en deux pesées. Examinons maintenant le cas général d'un trésor de n pièces.

Algorithme dans le cas général d'un trésor de n pièces (n est une donnée d'entrée de l'algorithme).

4. Dans le cas où n est une puissance de 3 ($n = 3^k$, k entier naturel) : écrire un algorithme itératif qui permet d'identifier, parmi n pièces, la pièce fautive qui est plus légère que les autres.
5. Expliquer comment votre algorithme fonctionne avec 27 pièces.
Aide : on devrait obtenir le résultat en 3 pesées.
6. Dans le cas d'un nombre quelconque de pièces, adapter l'algorithme précédent pour identifier la fautive pièce.
7. Afin de comparer avec la recherche dichotomique, remplir le tableau suivant, en indiquant, pour chaque méthode et pour différentes valeurs de n , le nombre maximal de pesées.

n	Dichotomie	Trichotomie
2		
3		
6		
8		
9		
16		
27		

Quelle est, d'après vous, la méthode la plus efficace en nombre de pesées ?

I.2. La question 6 est difficile pour les élèves. En particulier, le terme *adapter* n'est pas suffisamment précis. **Proposer, en les justifiant, une ou plusieurs aides que vous pourriez proposer aux élèves afin de leur permettre de répondre à cette question.**

(Répondre en 6 lignes maximum.)

I.3. **En transposant les notions de variants et d'invariants, vues pour l'algorithme de recherche dichotomique dans un tableau trié en classe de 1^{ère} NSI, donner à vos élèves un schéma de la preuve de correction de l'algorithme par trichotomie.**

(Répondre en 10 lignes maximum.)

Partie II. Mastermind : le joueur humain devine

Vous proposez à vos élèves de Terminale, en spécialité NSI, un projet qui réalise le jeu du Mastermind. Les règles du Mastermind sont fournies en annexe 1. Dans ce projet, les élèves travailleront sur deux situations :

- le joueur (humain) tente de découvrir une configuration de pions cachée par la machine (le joueur est alors appelé le décodeur et la machine le codificateur) ;
- le joueur (humain) cache une configuration de pions à la machine qui doit la découvrir (le joueur est alors appelé le codificateur et la machine le décodeur).

Dans cette partie II., on considère que le joueur est le décodeur et que la machine est le codificateur.

Dans la suite, on notera N le nombre de couleurs et P le nombre de pions cachés. On appellera *configuration* un ensemble de P pions dont les positions et les couleurs sont déterminées. C est le nombre maximal de configurations que le décodeur peut proposer.

Vous préparez un énoncé à distribuer aux élèves qui a pour but de les guider dans la réalisation de ce projet.

II.1. Parmi les structures de données de type liste, p-uplet et dictionnaire, vous choisissez la structure de données de liste pour représenter une configuration. **Donner les arguments qui seront présentés aux élèves pour justifier ce choix.**

II.2. **Donner la correction de l'énoncé, donné ci-dessous (partie en italique), qui sera proposé aux élèves.** Pour la question 2.d de l'énoncé, seules les lignes à compléter peuvent être données sur votre copie avec les numéros de lignes associés.

1. La configuration choisie par le codificateur est générée aléatoirement.

Écrire une fonction Python qui renvoie une configuration aléatoire selon une loi uniforme sur l'ensemble des configurations. L'en-tête de cette fonction sera le suivant :

```
def genere_configuration(n_couleurs,n_positions)
```

Vous pourrez utiliser la fonction `randint` de la bibliothèque `random` donnée en annexe 2 pour choisir aléatoirement une couleur, qui est représentée par un entier entre 1 et N .

Il est possible d'écrire ce programme de façon itérative et de façon récursive. Vous écrirez les deux versions.

2. Programmation, en Python, du jeu Mastermind quand le décodeur est le joueur humain

(a) *Écrire une fonction Python itérative calculant et retournant le nombre de pions bien placés entre deux configurations, c'est-à-dire les pions de même couleur et de même position entre les deux configurations (nombre de fiches noires). On l'appellera `n_bien_places` et son en-tête est :*

```
def n_bien_places(configuration_1,configuration_2)
```

- (b) Donner un algorithme qui calcule le nombre de pions communs entre deux configurations (l'ordre des pions dans la configuration n'a pas d'importance). Si nécessaire, vous pourrez utiliser une structure de données auxiliaire en expliquant votre choix. La fonction Python qui implante cet algorithme sera appelé `n_pions_communs` et son en-tête est :

```
def n_pions_communs(configuration_1, configuration_2)
```

L'implantation de cette fonction n'est pas demandée.

- (c) En déduire une fonction Python renvoyant le nombre de fiches blanches et de fiches noires qui sont déterminées lorsque deux configurations sont comparées. Pour rappel, le nombre de fiches noires correspond au nombre de pions étant bien placés (et donc de la même couleur) et le nombre de fiches blanches correspond au nombre de pions de la bonne couleur qui ne sont pas bien placés. On appellera la fonction `n_noirs_blancs` et son en-tête est :

```
def n_noirs_blancs(configuration_1, configuration_2)
```

- (d) Compléter le squelette de la fonction Python interactive, donnée ci-dessous, permettant d'effectuer une partie de Mastermind dans laquelle le codificateur est la machine et le décodeur est le joueur humain. Cette fonction comprend la génération d'une configuration à faire deviner, la saisie des propositions successives de configurations par l'utilisateur et les réponses correspondantes de la machine via les fiches blanches et noires. Les `"..."` données dans le squelette sont à compléter et peuvent représenter des variables, des expressions, des instructions ou des suites d'instructions.

```

1  def joueur_devine(n_couleurs=6, n_positions=4, n_essais=10):
2
3      configuration_cachee =
4          genere_configuration(n_couleurs, n_positions)
5      print("Taille de la configuration cachée: ", n_positions)
6
7      i = ...          # nombre d'essais déjà réalisés
8      trouve = ...    # variable booléenne qui indique si la
9                      # configuration a été trouvée
10
11     while ... :
12         ...
13         m = input("Essai "+str(i)+" : ")
14             # le joueur entre une configuration
15         ... # conversion de la configuration entrée
16             # dans le bon type
17         print("          "+str(...)+" bien placés - "
18             +str(...)+" mal placés")
19         trouve = ...
20
21     if trouve :
22         return "Bravo vous avez trouvé en "
23             + str(...) + " essais"
24     else :
25         return "Perdu vous avez épuisé vos " + str(...)
26             + " tentatives, bonne réponse = "
27             + str(configuration_cachee)

```

II.3. L'année précédente, vous aviez aussi proposé, en TP, à une classe de terminale, de programmer le jeu du Mastermind avec 5 couleurs, 5 positions et un nombre maximal de 10 essais pour le décodeur, mais vous ne leur aviez pas fourni l'énoncé précédent que vous avez préparé cette année. Un groupe d'élèves vous a rendu la fonction donnée en annexe 4. **Analyser cette production, en se référant aux numéros de ligne si nécessaire.**
(Répondre en 20 lignes maximum.)

II.4. **Proposer une remédiation pour ce groupe d'élèves sur les points qui vous paraissent importants.**
(Répondre en 10 lignes maximum.)

Partie III. Mastermind : la machine devine

Dans cette partie, il s'agit d'étudier le cas où le joueur (humain) joue le rôle du codificateur et la machine celui du décodeur.

Vous réfléchissez aux différentes questions et activités que vous aimeriez aborder avec vos élèves. Répondez aux questions suivantes.

III.1. Étude de l'ensemble de toutes les configurations

III.1.a. **Donner la formule du nombre de configurations possibles en fonction des paramètres N et P du jeu.**

III.1.b. **Avec les valeurs habituelles pour le jeu, qui sont $N = 6$ et $P = 4$ pour les plus jeunes joueurs et joueuses, et dans la version classique $N = 8$ et $P = 5$, combien y a-t-il de configurations possibles?** Vous pourrez vous aider de la table de calculs donnée en annexe 3.

III.1.c. **Est-il envisageable et raisonnable de créer et de mémoriser toutes les configurations dans ces deux cas de figures?**
Quelle comparaison pourriez-vous proposer à vos élèves pour qu'ils puissent arriver à la même conclusion que vous? Vous pourrez supposer qu'un entier est codé sur 64 bits.

III.2. Vous avez trouvé sur le Web un projet, traitant du Mastermind, déjà réalisé. Pour savoir si vous pouvez vous en inspirer, vous l'analysez. **Que fait la fonction Python suivante? Prouver qu'elle n'omet aucune configuration.**

```
def mystere(n_couleurs = 6, n_positions = 4) :
    if ( n_positions == 0 ) :
        return [[]]
    else :
        l = []
        for element in mystere(n_couleurs, n_positions-1):
            for couleur in range(1, n_couleurs+1) :
                l.append(element + [couleur])
        return l
```

Comme cela a été fait dans ce projet, vous décidez vous aussi d'utiliser une liste pour stocker toutes les configurations possibles.

III.3. Donner un plan de cours, de niveau NSI Terminale, sur les structures de données linéaires, comme les listes, piles et files, en explicitant les exemples utilisés comme illustrations. Un extrait du programme concernant ce sujet est donné dans la table 1.

(Répondre en 10 lignes maximum.)

Contenu	Capacités attendues	Commentaires
Listes, piles, files : structures linéaires	Distinguer des structures par le jeu des méthodes qui les caractérisent. Choisir une structure de données adaptée à la situation à modéliser.	On distingue les modes FIFO (first in first out) et LIFO (last in first out) des piles et des files. Les listes n'existent pas de façon native en Python.

TABLE 1 – Extrait du programme de NSI Terminale - Structures de données.

III.4. Vous réfléchissez aux différents éléments à donner et aux programmes à demander aux élèves lorsque la machine tente de deviner la configuration choisie par le codificateur.

III.4.a. Vous supposez que la machine a proposé une configuration, nommée `tentative`, et a reçu du codificateur, en retour, le nombre de fiches noires, noté `n_bien_places`, et le nombre de fiches blanches, noté `n_mal_places`.

Vous comptez demander à vos élèves un algorithme qui commence avec l'ensemble des configurations qui étaient encore candidates avant la proposition de la configuration `tentative` (ensemble qui sera appelé `configurations_possibles`) et qui le modifie pour qu'il ne contienne plus que l'ensemble des configurations qui pourront être candidates au tour suivant, une fois les fiches noires et blanches reçues du codificateur suite à la proposition de `tentative`. **Écrire cet algorithme pour pouvoir donner un corrigé à vos élèves.** La fonction Python correspondante s'appellera `reduction_configurations_possibles`. Il ne vous est pas demandé d'écrire cette fonction, mais seulement de **donner son en-tête avec ses paramètres d'entrée**.

III.4.b. **Quel(s) argument(s) donneriez-vous à vos élèves pour qu'ils choisissent la liste, parmi les structures linéaires de type liste, pile et file, afin de stocker l'ensemble des configurations possibles ?**

III.4.c. **Quel argument donneriez-vous à vos élèves pour qu'ils comprennent que n'importe quelle configuration peut être choisie dans l'ensemble `configurations_possibles` pour le tour suivant ?**

III.4.d. **Écrire le corrigé de la fonction Python interactive, demandée à vos élèves, permettant d'effectuer une partie où c'est la machine qui devine la configuration choisie par le joueur. Ce programme comprend la génération de toutes les configurations possibles par la machine, les propositions successives de la machine et les réponses correspondantes du joueur.**

Voici son en-tête :

```
def machine_devine(n_couleurs=6,n_positions=4,n_essais=10)
```

Partie IV. Mastermind : évaluation et analyse d'un projet

Vous donnez le projet de Mastermind à vos élèves. Vous les guidez avec les différentes étapes que vous avez établies dans les parties II. et III. Vous avez réservé 20h pour travailler sur ce projet avec vos élèves. À la fin du projet, les élèves devront rendre un programme qui permet d'effectuer une partie, avec le joueur (humain) qui peut être codificateur ou décodeur.

IV.1. Proposer un barème qui vous permettra d'évaluer le travail rendu par les élèves. Expliquer les différents éléments que vous comptez évaluer ainsi que le nombre de points attribués pour chacun de ces éléments.

(Répondre en 15 lignes maximum.)

Une élève vous a rendu un projet dont un extrait figure en annexe 5. Elle est allée beaucoup plus loin que ce qui était attendu sur ce projet. Notamment, elle a fait jouer la machine contre elle-même et elle a estimé expérimentalement le nombre moyen d'essais effectués par la machine pour trouver la configuration cachée. Pour cela, la machine a généré des configurations aléatoires avec un nombre de couleurs N et une taille P fixés et l'élève a compté le nombre d'essais effectués par la machine pour trouver chaque configuration. Elle a limité à 10 le nombre maximal d'essais.

IV.2. Que pensez-vous de son approche qui consiste à baser son estimation du nombre moyen d'essais effectués sur plusieurs configurations aléatoires ? Justifier votre réponse.

IV.3. Que pensez-vous de sa remarque : "... plus le nombre de boules et le nombre de couleurs sont élevés, plus le nombre de simulations est bas..." ? Justifier votre réponse.

IV.4. Dans la version pour jeunes joueurs du jeu, $N = 6$ et $P = 4$, pensez-vous qu'il est raisonnable de fixer la limite du nombre d'essais à 10 ? Justifier votre réponse à partir des valeurs obtenues par l'élève sur le nombre moyen d'essais.

IV.5. Quel indicateur l'élève aurait-elle pu mesurer expérimentalement pour affiner la limite sur le nombre d'essais raisonnables à fixer ?

IV.6. Qu'est-ce que l'élève aurait pu mesurer pour analyser, expérimentalement, le coût de son algorithme, en plus du nombre d'essais effectués ?

IV.7. Vous aidez l'élève à réfléchir à la complexité du problème de recherche d'une configuration cachée. Pour cela, vous lui décrivez un algorithme qui trouve la configuration cachée en $N + \frac{P(P-1)}{2}$ essais pour N couleurs et P positions avec $N > P$. Décrire cet algorithme. Quel est le nombre d'essais proposés par cet algorithme dans le cas du jeu pour jeunes joueurs et joueuses : $N = 6$ et $P = 4$? Et dans le cas classique $N = 8$ et $P = 5$?

IV.8. Que devrait en conclure l'élève sur le problème de recherche d'une configuration cachée ? Justifier votre réponse.

IV.9. En vous inspirant de l'idée de la partie I., montrer que le problème de recherche pour N couleurs et P positions ne peut pas être résolu, pour le pire cas, en moins de $\log_b N^P$ où b est une base que l'on calculera.

Indication : calculer le nombre de réponses différentes possibles, en termes de fiches noires et blanches, pour un essai.

Partie V. Enjeu sociétal

Vous êtes sensible à la protection des données personnelles, sujet qui peut être abordé, entre autres, dans les thématiques Web et Réseaux sociaux du programme SNT. Vous avez l'intention de consacrer 1h30, en classe de Seconde, lors de l'enseignement de SNT, sur les questions d'anonymat et de ré-identification et plus généralement sur les risques liés aux données individuelles que l'on laisse sur le Web et en particulier sur les réseaux sociaux. Des extraits du programme de SNT sur ces thématiques sont donnés en annexe 6. Vous avez aussi trouvé un article, donné en annexe 6, qui traite du problème de la ré-identification et que vous allez exploiter pour votre cours.

V.1. En quoi le problème de l'anonymat et de la ré-identification est-il similaire au principe du Mastermind, projet que vous avez travaillé avec vos élèves de Terminale ? Expliquer comment vous pourriez utiliser le Mastermind pour illustrer les principes d'anonymat et de ré-identification à vos élèves de Seconde.

V.2. Donner 3 illustrations que vous pourriez présenter aux élèves afin qu'ils prennent conscience de toutes les informations personnelles qu'ils peuvent être amenés à donner et qui peuvent permettre leur ré-identification.

V.3. Citer 2 contextes où le partage de données anonymisées est d'intérêt public.

V.4. Citer 2 situations où la ré-identification est problématique.

V.5. Citer 3 mini-activités sur machine qui pourraient aider les élèves à comprendre quelles informations personnelles sont stockées et comment ils peuvent essayer de réguler les traces qu'ils laissent dans leurs usages quotidiens.

V.6. Donner le plan de cours d'1h30 que vous projetez de faire sur ce sujet. Vous décrierez les différentes notions que vous comptez aborder et les illustrations associées le cas échéant, ainsi que les activités que vous comptez réaliser avec les élèves et l'ordre dans lequel vous comptez aborder ces notions et ces activités.

(Répondre en 20 lignes maximum.)

V.7. L'évaluation des acquis sur ce cours se fera par QCM. Proposer deux questions, avec 4 choix chacune, qui testeront, pour l'une, les connaissances acquises et pour l'autre, le savoir-faire (comme le raisonnement et la mise en œuvre par exemple). Justifier le choix des questions. Pour chaque question, justifier le choix des 4 réponses proposées. Vous indiquerez aussi si les réponses sont justes ou fausses.

Annexe 2 : Bibliothèque random

Bibliothèque random de Python

Voici quelques explications pour utiliser la fonction `randint` de la bibliothèque `random` de Python.

Tirer aléatoirement un nombre entier

Pour tirer aléatoirement un nombre entier selon une loi uniforme, on peut utiliser la fonction `randint`. Cette fonction prend deux paramètres en entrée : une valeur minimale et une valeur maximale.

Voici un exemple pour tirer des entiers entre 1 et 5. Le premier paramètre doit être inférieur au second.

```
import random
print random.randint(1, 5)
```

Le résultat sera soit 1, soit 2, soit 3, soit 4 ou 5.

Annexe 3 : Table de calculs

La table suivante donne la valeur de x^y pour différentes valeurs de x et de y .

	$y = 2$	$y = 3$	$y = 4$	$y = 5$	$y = 6$	$y = 7$	$y = 8$	$y = 9$
$x = 2$	4	8	16	32	64	128	256	512
$x = 3$	9	27	81	243	729	2187	6561	19683
$x = 4$	16	64	256	1024	4096	16384	65536	262144
$x = 5$	25	125	625	3125	15625	78125	390625	1953125
$x = 6$	36	216	1296	7776	46656	279936	1679616	10077696
$x = 7$	49	343	2401	16807	117649	823543	5764801	40353607
$x = 8$	64	512	4096	32768	262144	2097152	16777216	134217728

Annexe 4 : Production d'élèves

Production d'élèves – partie II., question II.3

```
1 def verification():
2     # definit le nombre de couleurs justes
3     # et de couleurs qui ne sont pas dans la combinaison a trouver
4     bon=0
5     mauvais=0
6     tour=tour+1
7     if a==aa:
8         bon=bon+1
9     if aa!=a and aa!=b and aa!=c and aa!=d and aa!=e:
10        mauvais=mauvais+1
11    if b==bb:
12        bon=bon+1
13    if bb!=b and bb!=a and bb!=c and bb!=d and bb!=e:
14        mauvais=mauvais+1
15    if c==cc:
16        bon=bon+1
17    if cc!=c and cc!=b and cc!=a and cc!=d and cc!=e:
18        mauvais=mauvais+1
19    if d==dd:
20        bon=bon+1
21    if dd!=d and dd!=b and dd!=c and dd!=a and dd!=e:
22        mauvais=mauvais+1
23    if e==ee:
24        bon=bon+1
25    if ee!=e and ee!=b and ee!=c and ee!=d and ee!=a:
26        mauvais=mauvais+1
27    if a==aa and b==bb and c==cc and d==dd and e==ee:
28        # Si la combinaison est la même que celle é trouver ,
29        # ouvre une fenêtre de victoire
30        fen_gagne=Toplevel()
31        fen_gagne.title("Mastermind")
32        text=Label(fen_gagne, text="Vous avez gagné !
33                    Félicitations !")
34        text.pack()
35        global fen_gagne
36    if tour==10 :
37        # Si au 10eme tour , la combinaison n'a pas encore ete
38        #                               trouvée, défaite
39        if aa!=a or bb!=b or cc!=c or dd!=d or ee!=e :
40            fen_perdu=Toplevel()
41            fen_perdu.title("Mastermind")
42            text=Label(fen_perdu, text=
43                    "C est la 10e tentative sans succès,
44                    vous avez donc perdu la partie")
45            text.pack()
46            global fen_perdu
47    bon=str(bon)
48    mauvais=str(mauvais)
49    global tour, bon, mauvais
```

Annexe 5 : Analyse d'un projet d'élève sur le Mastermind

Analyse d'un projet d'élève sur le Mastermind – partie IV.

Extraits d'un projet d'élève sur le Mastermind (accessible à <https://issuu.com/wile/docs/mastermind>).

Note : dans cette production, la « version classique » correspond à la version pour les plus jeunes joueurs du Mastermind, avec 4 positions et 6 couleurs et une boule correspond à un pion.

4. Analyse des résultats

4.1. Le nombre de coups pour gagner

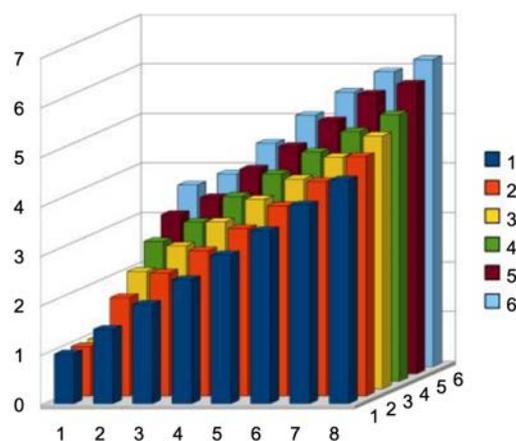
Grâce aux nombreuses simulations, j'ai pu découvrir le nombre de coups qu'il faut en moyenne pour découvrir la combinaison cachée en fonction du nombre de couleurs et du nombre de boules. J'ai effectué de 100 à 10'000 simulations par cas, cela dépend de la complexité de l'opération. Ainsi, plus le nombre de boules et le nombre de couleurs sont élevés, plus le nombre de simulations est bas, car sinon, cela prendrait trop de temps. Les valeurs sont donc moins précises pour ces cas.

Moyenne du nombre de coups en fonction du nombre de boules et de couleurs

	1 boule	2 boules	3 boules	4 boules	5 boules	6 boules
1 couleur	1.000	1.000	1.000	1.000	1.000	1.000
2 couleurs	1.503	1.996	2.373	2.822	3.216	3.663
3 couleurs	2.006	2.485	2.884	3.214	3.562	3.893
4 couleurs	2.499	2.938	3.371	3.740	4.135	4.507
5 couleurs	3.008	3.387	3.825	4.192	4.596	5.070
6 couleurs	3.495	3.844	4.238	4.631	5.117	5.540
7 couleurs	4.014	4.338	4.680	5.043	5.650	5.950
8 couleurs	4.513	4.841	5.109	5.388	5.850	6.200

Rouge : Mastermind classique

Graphique correspondant au tableau ci-dessus



axe Z (verticalement) : nombre de coups

axe Y (en profondeur) : nombre de boules

axe X (horizontalement de gauche à droite) : nombre de couleurs

Annexe 6

Documents pour la partie V.

Cette annexe comprend un extrait du programme de SNT en Seconde concernant les thématiques du Web et des réseaux sociaux, ainsi qu'un article tiré du site *The Conversation*.

Extrait du programme de SNT en Seconde concernant la thématique du Web.

Impacts sur les pratiques humaines

Dans l'histoire de la communication, le Web est une révolution : il a ouvert à tous la possibilité et le droit de publier ; il permet une coopération d'une nature nouvelle entre individus et entre organisations : commerce en ligne, création et distribution de logiciels libres multi-auteurs, création d'encyclopédies mises à jour en permanence, etc. ; il devient universel pour communiquer avec les objets connectés.

Le Web permet aussi de diffuser toutes sortes d'informations dont ni la qualité, ni la pertinence, ni la véracité ne sont garanties et dont la vérification des sources n'est pas toujours facile. Il conserve des informations, parfois personnelles, accessibles partout sur de longues durées sans qu'il soit facile de les effacer, ce qui pose la question du droit à l'oubli. Il permet une exploitation de ses données, dont les conséquences sociétales sont encore difficiles à estimer : recommandation à des fins commerciales, bulles informationnelles, etc. En particulier, des moteurs de recherche permettent à certains sites d'acquérir de la visibilité sur la première page des résultats de recherche en achetant de la publicité qui apparaîtra parmi les liens promotionnels.

Extrait du programme de SNT en Seconde concernant la thématique des réseaux sociaux.

Les données et l'information

Les différents réseaux sociaux permettent l'échange d'informations de natures différentes : textes, photos, vidéos. Certains limitent strictement la taille des informations, d'autres autorisent la publication, mais de façon limitée dans le temps. Certains permettent l'adjonction d'applications tierces (plug-ins) qui peuvent ajouter des fonctionnalités supplémentaires.

Toutes les applications de réseautage social utilisent d'importantes bases de données qui gèrent leurs utilisateurs, l'ensemble des données qu'ils partagent, mais aussi celles qu'ils consentent à fournir (sans toujours le savoir), y compris sur leur vie personnelle.

Article tiré du site *The Conversation*

Seules 3 pages de cet article sont données (la dernière page n'a pas été reproduite, elle contenait des suggestions de lecture et ne concernait pas le sujet que vous devez traiter).

Cet article comporte une erreur de frappe : il faut lire **font** à la place de **sont** dans la phrase *En réponse, les compagnies et organismes qui les collectent affirment souvent qu'elles le sont de manière « anonyme »*.

THE CONVERSATION

L'expertise universitaire, l'exigence journalistique



Dans le métro. Photo by Martin Adams on Unsplash

Données anonymes... bien trop faciles à identifier

17 septembre 2019, 21:01 CEST

Téléphones, ordinateurs, cartes de crédit, dossiers médicaux, montres connectées, ou encore assistants virtuels : chaque instant de nos vies – en ligne et hors ligne – produit des données personnelles, collectées et partagées à grande échelle. Nos comportements, nos modes de vie, s’y lisent facilement. Mais faut-il s’en inquiéter ? Après tout, ces données qui nous révèlent sont souvent anonymisées par les organismes qui les collectent. C’est du moins ce que l’on peut lire sur leurs sites. Leur travail est-il efficace ? Et les données anonymes le sont-elles vraiment ? Dans notre dernier article publié dans la revue *Nature Communications*, nous développons une méthode mathématique qui montre que c’est loin d’être acquis. Elle a pu nous amener à réidentifier des individus parmi des bases de données anonymes et fortement échantillonnées, remettant en question les outils utilisés actuellement pour partager les données personnelles à travers le monde.

Matière première

D’abord, quelques ordres de grandeur. Ces dix dernières années, nos données personnelles ont été collectées à une vitesse inégalée : 90 % de celles circulant sur Internet ont été créées il y a moins de deux ans ! Objets connectés, informations médicales ou financières, réseaux sociaux, ces données sont

Auteur



Luc Rocher

Doctorant, ingénierie mathématique,
Université catholique de Louvain

15 / 17

la matière première de l'économie numérique comme de la recherche scientifique moderne. Mais, très vite, on a vu apparaître certaines dérives. Notamment les atteintes à la vie privée qui se sont multipliées. Témoin, parmi de nombreuses affaires, le **scandale Cambridge Analytica**... Depuis, 80 % des Européen-ne-s estiment avoir perdu le contrôle sur leurs données.

En réponse, les compagnies et organismes qui les collectent affirment souvent qu'elles le sont de manière « anonyme ». Par exemple, la société Transport for London (TfL), en charge du métro londonien, a entrepris de surveiller les déplacements des passagers sur le réseau via les signaux wifi « anonymes » de leurs téléphones portables. En Belgique, plus de 15 hôpitaux revendent les données confidentielles de leurs patients à une multinationale, Quintiles IMS, sous couvert d'anonymat. Enfin, en France, Orange et SFR ont revendu des données de géolocalisation en temps réel ou en différé, données là encore « anonymisées ».

Point intéressant, une donnée anonyme n'est plus considérée comme donnée personnelle. Elle échappe donc aux régimes de protection comme le RGPD en Europe. Partager des données personnelles anonymisées ne nécessite donc plus le consentement des participant-e-s... Puisqu'ils et elles sont anonymes !

Ré-identification

Or, des chercheur-e-s et journalistes ont depuis longtemps montré que certaines données anonymes peuvent être ré-identifiées. Dans les années 1990, Latanya Sweeney avait pu ré-identifier les données médicales de William Weld (alors gouverneur du Massachusetts), sur base de son code postal, sa date de naissance et son genre. Deux journalistes allemands ont récemment ré-identifié l'historique de navigation d'un juge et d'un député, retrouvant leurs préférences sexuelles et leurs traitements médicaux dans des données anonymes obtenues en se faisant passer pour des acheteurs potentiels. Et, aux États-Unis, les dossiers fiscaux du président américain Trump ont pu lui être ré-attribués par le *New York Times* en utilisant des données anonymes publiées par le fisc américain, l'IRS.

Compagnies et gouvernements minimisent souvent ces ré-identifications. Leur ligne de défense : parmi des petites bases de données, toujours incomplètes, personne ne saura jamais si une ré-identification est correcte ou non et si des chercheur-e-s ou journalistes ont vraiment ré-identifié la bonne personne.

Cela implique que l'organisme collecteur fasse un travail dit d'*échantillonnage* sur la base de données. Ainsi, l'autorité de protection des données australienne [OAIC], suggère dans son **guide de dés-identification** que l'échantillonnage augmente « l'incertitude qu'une personne particulière fasse réellement partie d'une base de données anonyme ». Prenons un exemple pour expliquer cela. Admettons que votre employeur retrouve des données vous correspondant dans un échantillon de 10 000 patients, soit 1 % d'une large base de données médicales. Ces données – comprenant par exemple votre lieu et date de naissance, genre, statut marital, etc. – pourraient bien appartenir à une autre personne qui partage ces caractéristiques. Car cette base de données de 10 000 personnes ne représente que 0,015 % de la population française. Et ces données ré-identifiées pourraient



Traitement d'échantillons viraux. Les données personnelles de santé sont parmi les plus sensibles. James Gathany/CDC

correspondre à n'importe quelle autre personne parmi les 99,985 % autres Français-e-s.

Échantillonner (partager par exemple 1 % d'une base de données) est ainsi une technique largement utilisée. Réduire la taille des données partagées permet de justifier que ces données sont anonymes, car personne ne pourra jamais prouver qu'une ré-identification est correcte.

Un algorithme qui remet en question l'anonymat

Le problème ? Nos travaux démontrent au contraire qu'un algorithme peut apprendre à estimer, avec grande précision, si des données réidentifiées appartiennent bien à la bonne personne ou non.

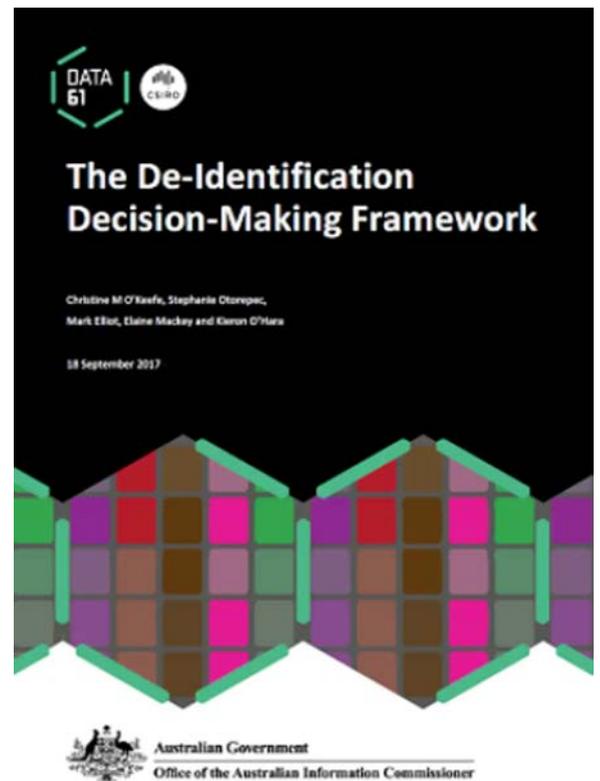
Il y a bien entendu, si c'est en France, de nombreux hommes trentenaires, habitant à Paris. Si je retrouve un seul homme de 30 ans parmi les données anonymes de 1 000 personnes, collectées et revendues par un cabinet d'assurance parisien, il y a peu de chance qu'elles correspondent à mon voisin Émeric. Les données correspondant à ces trois attributs (homme, 30 ans, habitant à Paris) seront sans doute celles d'un autre Français.

Mais au fur et à mesure que ces données s'enrichissent, qu'on apprend davantage de caractéristiques, il devient illusoire qu'une seconde personne ait les mêmes caractéristiques. Il y a ainsi sans doute un seul homme à Paris, né le 5 janvier 1989, roulant en vélo électrique et habitant avec ses deux enfants (deux filles) et un berger allemand : mon voisin Émeric.

Après avoir « appris » quelles caractéristiques rendent les individus uniques, notre algorithme génère des populations synthétiques pour estimer si un individu peut se démarquer parmi des milliards de personnes. Le modèle développé permettrait par exemple aux journalistes du New York Times de savoir à coup sûr si les dossiers identifiés appartenaient vraiment à Donald Trump.

Nos résultats montrent que 99,98 % des Américains seraient correctement ré-identifiés dans n'importe quelle base de données en utilisant 15 attributs démographiques. Les chiffres sont similaires à travers le monde (16 attributs en ajoutant la nationalité). Une quinzaine de caractéristiques qui suffisent à identifier un individu, ce n'est hélas pas beaucoup. Le « data broker » Axiom, un courtier de données qui achète et qui revend nos données personnelles dans 60 pays, possède par exemple jusqu'à 5,000 attributs par personne.

Nos travaux remettent ainsi en question les pratiques actuelles utilisées pour dés-identifier des données personnelles. Cela interroge sur les limites de l'anonymisation : utiliser ainsi ces données protège-t-il toujours notre vie privée ? Alors que les standards d'anonymisation sont en passe d'être redéfinis par les pouvoirs publics, au niveau national et au sein de l'Union européenne, il est crucial pour ces standards d'être rigoureux, de promouvoir de meilleures méthodes de partage des données, et de prendre en compte tout risque futur. C'est à la fois important pour nos vies privées, pour la croissance de l'économie numérique et pour le dynamisme de la recherche scientifique.



Un guide pour protéger les données en Australie. Australian Government, CC BY