Option Informatique en Spé MP et MP*
Devoir à rendre après les vacances de la Toussaint
Distance d'édition, plus long sous-mot commun

Les questions marquées ⋆⋆⋆ sont délicates ou subtiles (à mon avis). La première partie est fortement indépendante des suivantes, qui n'utilisent que marginalement les résultats des questions 1 à 4.

L'emploi de références est interdit dans les programmes en Caml.

1 Sous-mots

Mark how one string, sweet husband to another,
Strikes each in each by mutual ordering...

Shakespeare — Sonnet 8

▶ Σ désigne un alphabet contenant au moins deux lettres a et b. Un mot x est un *sous-mot* d'un mot y (ce que l'on notera $x \sqsubseteq y$) si, notant $m = |x|$ et $n = |y|$, il existe une injection croissante s de $[\![1, m]\!]$ dans $[\![1, n]\!]$ telle que $x_i = y_{s(i)}$ pour tout $i \in [\![1, m]\!]$. Ceci revient à dire que x peut se déduire de y en *supprimant* certains caractères ; ou, inversement, que y peut se déduire de x en *insérant* des caractères. Ainsi, *corne* est un sous-mot de *recouvrante*.

▶ On notera que, si x est facteur de y, alors x est un sous-mot de y ; mais la réciproque n'est pas vraie.

▶ On note $|X|$ le cardinal de l'ensemble fini X, et $\lg x$ le logarithme en base 2 du réel $x > 0$.

Question 1 • Citez quelques sous-mots remarquables du mot *découvrable*.

Question 2 • Justifiez rapidement : la relation $\sqsubseteq$ est un ordre, compatible avec la concaténation : si $x \sqsubseteq y$, alors $xz \sqsubseteq yz$ et $zx \sqsubseteq zy$.

Question 3 • Rédigez en Caml une fonction :

```
est_sous_mot : string -> string -> bool
```

spécifiée comme suit : `est_sous_mot x y` dit si x est un sous-mot de y. Vous justifierez en toute rigueur la validité de cette fonction.

Question 4 • Montrez qu'avec cette fonction, le coût de `est_sous_mot x y` (exprimé en nombre de comparaisons de caractères) est au plus égal à n.

▶ Soit $w \in \Sigma^*$; on note $S(w)$ l'ensemble des sous-mots de w et $s(w) = \big|S(w)\big|$.

Question 5 • Soit $n \in \mathbb{N}$; calculez $\min\{s(w) \mid w \in \Sigma^n\}$ et précisez les mots pour lesquels ce minimum est atteint. Justifiez la majoration $s(w) \leqslant 2^{|w|}$.

▶ Dans la suite de cette partie, on suppose que Σ se réduit à deux lettres, donc $\Sigma = \{a, b\}$. On se propose d'expliciter $\max\{s(w) \mid w \in \Sigma^n\}$, en précisant les mots pour lesquels ce maximum est atteint.

▶ Soit $P \in \mathbb{R}[X]$; pour $k \in \mathbb{N}$, on note $[X^k]P$ le coefficient de X^k dans P, ainsi $P = \sum_{k\in\mathbb{N}} \big([X^k]P\big)X^k$. À un langage fini L, on associe le polynôme P_L défini par :

$$P_L = \sum_{k\in\mathbb{N}} \big|L \cap \Sigma^k\big| X^k$$

$[X^k]P_L$ est donc le nombre de mots de longueur k qui appartiennent à L.

▶ Soit $w \in \Sigma^*$; on note $P_w = P_{S(w)}$, et $P^a_w = P_{S(w)\cap a\Sigma^*}$; ainsi $[X^k]P^a_w$ est le nombre de sous-mots de w qui commencent par la lettre a et dont la longueur est k.

▶ Soient U et V deux polynômes ; on note $U \preceq V$ si $[X^k]U \leqslant [X^k]V$ pour tout $k \in \mathbb{N}$; on note $U \prec V$ si $U \preceq V$ et $U \neq V$. Clairement, $\preceq$ est un ordre partiel, compatible avec l'addition.

Question 6 • Explicitez P_{aaaba}.

Question 7 • Établissez la relation $P_{xy} \preceq P_x P_y$.

Question 8 • Justifiez $P^b_{aw} = P^b_w$, puis $P^a_{aw} = X(P^a_w + P^b_w + 1)$. Combien vaut P^a_ε ?

▶ On définit une suite $(Q_n)_{n\in\mathbb{N}}$ de polynômes par les formules $Q_0 = 0$, $Q_1 = X$ et $Q_{n+2} = X(Q_{n+1} + Q_n + 1)$. On définit également deux suites $(\alpha_n)_{n\in\mathbb{N}}$ et $(\beta_n)_{n\in\mathbb{N}}$ de mots par les formules $\alpha_0 = \beta_0 = \varepsilon$, $\alpha_{n+1} = a\beta_n$ et $\beta_{n+1} = b\alpha_n$. Ainsi $|\alpha_n| = |\beta_n| = n$, $\alpha_n = ababa\ldots$ et $\beta_n = babab\ldots$; plus précisément, $\alpha_{2p} = (ab)^p$, $\alpha_{2p+1} = (ab)^p a$, $\beta_{2p} = (ba)^p$ et $\beta_{2p+1} = (ba)^p b$.

Question 9 • Justifiez : $P^a_{\alpha_n} = P^b_{\beta_n} = P^a_{\beta_{n+1}} = P^b_{\alpha_{n+1}} = Q_n$.

Question 10 • La *suite de Fibonacci* est définie par les relations $F_0 = 0$, $F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$. Explicitez $s(\alpha_n)$ en fonction d'un ou plusieurs termes de cette suite.

Question 11 • Justifiez la relation $Q_n \prec Q_{n+1}$.

Question 12 ⋆⋆⋆ • Soit u un mot de longueur n, distinct de α_n et β_n. Montrez qu'il existe un mot v de même longueur, et vérifiant $P_u \prec P_v$.

Question 13 • Et maintenant, concluez !

2 Distance de Levenshtein

If the dull substance of my flesh were thought,
Injurious distance should not stop my way...

Shakespeare — Sonnet 44

▶ Dans tout le problème, x et y sont deux mots sur Σ, de longueurs respectives m et n. On note $d(x, y)$ le nombre minimal d'insertions et/ou de suppressions de caractères nécessaires pour transformer x en y.

Question 14 • Justifiez le fait que d est une distance ; établissez l'encadrement $|m - n| \leqslant d(x, y) \leqslant m + n$, caractérisez les couples (x, y) pour lesquels le majorant est atteint, puis ceux pour lesquels le minorant est atteint.

▶ Pour $i \in [\![0, m]\!]$, on note $x[1..i]$ le préfixe de x de longueur i ; bien entendu, $x[1..0] = \varepsilon$. On définit de même $y[1..j]$ pour $j \in [\![0, n]\!]$. On note $D(i, j)$ la distance des mots $x[1..i]$ et $y[1..j]$.

Question 15 • Combien valent respectivement $D(i, 0)$ et $D(0, j)$?

Question 16 • Pour $i \in [\![1, m]\!]$ et $j \in [\![1, n]\!]$, on note $\alpha_{i,j}$ le nombre égal à 2 si $x_i \neq y_j$, à 0 dans le cas contraire. Justifiez la relation :

$$D(i, j) = \min\big(D(i-1, j-1) + \alpha_{i,j}, D(i-1, j) + 1, D(i, j-1) + 1\big)$$

Question 17 • En déduire une méthode de calcul de $d(x, y)$; vous ferez en sorte que le nombre d'opérations et l'espace mémoire requis soient des $\mathcal{O}(mn)$. Mettez en œuvre cette méthode en Caml.

Question 18 • Expliquez comment faire pour que l'espace mémoire requis soit un $\mathcal{O}\big(\min(m, n)\big)$. Rédigez en Caml une fonction exploitant cette idée.

Question 19 • Calculez $d(acbcab, bccaacba)$.

3 Script d'édition

The strings, my lord, are false.

Shakespeare — Jules César

▶ Soient σ, ι et ρ trois lettres n'appartenant pas à Σ. Un *script d'édition* est un mot t sur l'alphabet $\Sigma \cup \{\sigma, \iota, \rho\}$; σ indiquera une *suppression*, ι une *insertion* et ρ une recopie. Le résultat de l'action d'un script t sur un mot x est le mot $\varphi(t, x, \varepsilon)$, où φ est défini par les formules suivantes, dans lesquelles $a \in \Sigma$:

$$\begin{aligned}
\varphi(\varepsilon, \varepsilon, y) &= y \\
\varphi(\sigma t, ax, y) &= \varphi(t, x, y) \\
\varphi(\iota a t, x, y) &= \varphi(t, x, ya) \\
\varphi(\rho t, ax, y) &= \varphi(t, x, ya)
\end{aligned}$$

Un script d'édition est *valide* pour un mot x s'il est possible de l'appliquer complètement ; par exemple, $\rho\sigma\iota d\rho$ est un script valide pour *bca*, son application donne le mot *bda*. En revanche, les scripts $\rho\rho\sigma\sigma$, $\rho\sigma\iota$ et $\rho\iota d$ ne sont pas valides pour le mot *bca*.

Question 20 • Rédigez en Caml une fonction `phi` traduisant la définition de φ.

Question 21 • Dans quelle(s) situation(s) le calcul se bloque-t-il ?

Question 22 • L'ensemble $\mathcal{S}_x$ des scripts valides pour un mot x donné est-il fini ? Est-ce un langage rationnel ?

Question 23 • L'ensemble $\mathcal{R}_x$ des mots que l'on peut obtenir en appliquant à un mot x donné un script valide est-il rationnel ?

Question 24 • Montrez que, si l'on dispose du tableau D construit pour calculer $d(x, y)$, on peut déterminer un script d'édition transformant x en y et dont le nombre d'opérations (insertions et/ou suppressions) est minimal.

Question 25 • Appliquez le résultat précédent à la détermination d'un script d'édition minimal transformant $x = acbcab$ en $y = bccaacba$.

Question 26 • Étudiez les modifications apportées par l'introduction d'une opération supplémentaire, la *substitution* d'un caractère à un autre. Quelle autre opération peut-il être intéressant d'introduire ? Discutez également l'utilité d'une fonction de *pondération*, qui au couple de lettres (a, b) associe le coût du remplacement de a par b.

4 Plus long sous-mot commun

When such strings jar, what hope of harmony ?
Shakespeare — Deuxième Henry VI

Question 27 • Justifiez l'existence d'un plus long sous-mot z commun à x et y. Est-il unique ?

Question 28 • Soit z un plus long sous-mot commun à x et y. Justifiez la relation $|x| + |y| = d(x, y) + 2|z|$.

▶ La *méthode de la force brutale* consiste à déterminer l'ensemble des sous-mots de x et celui des sous-mots de y, à construire leur intersection, et à calculer la longueur maximale d'un élément de cette intersection.

Question 29 • Expliquez comment mettre en œuvre cette méthode pour obtenir un coût en $\mathcal{O}(mn2^n)$.

5 Algorithme de Wagner et Fischer

. . . by fair sequence and succession. . .
Shakespeare — Richard II

▶ Pour $i \in [\![0, m]\!]$ et $j \in [\![0, n]\!]$, on note $L(i, j)$ la longueur d'un plus long sous-mot commun à $x[1..i]$ et $y[1..j]$.

Question 30 • Donnez des formules permettant de remplir, de proche en proche, le tableau L.

Question 31 • Appliquez ces formules à la construction du tableau L associé aux deux mots $x = acbcab$ et $y = bccaacba$.

Question 32 • Expliquez comment l'on peut, en lisant le tableau L, déterminer un plus long sous-mot commun à x et y.

Question 33 • Montrez que la détermination d'un tel mot revient à la détermination d'un plus court chemin dans un graphe que l'on explicitera.

Question 34 • En déduire un plus long sous-mot commun à $x = acbcab$ et $y = bccaacba$.

6 Algorithme de Hirschberg

Who, every word by all my wit being scann'd,
Want wit in all one word to understand.
Shakespeare — La comédie des erreurs

▶ L'algorithme précédent a le défaut de nécessiter un espace mémoire proportionnel à mn. Hirschberg a proposé en 1975 un algorithme ne demandant qu'un espace mémoire proportionnel à $\min(m,n)$; cet algorithme a été utilisé en 1975 par Hunt et McIlroy dans le programme `diff` d'Unix.

▶ On note $L^*(i,j)$ la longueur d'un plus long sous-mot commun à $x[i+1..m]$ et $y[j+1..n]$ (ces deux dernières notations étant claires), et on définit :

$$M(i) = \max_{0 \leqslant j \leqslant n} \big(L(i,j) + L^*(i,j)\big)$$

On suppose $m \geqslant n$ pour fixer les idées.

Question 35 • Montrez que $M(i) = L(m,n)$ pour tout $i \in [\![0,m]\!]$.

Question 36 ⋆⋆⋆ • En prenant $i = \lfloor m/2 \rfloor$, montrez que l'on atteint bien le résultat annoncé.

7 Algorithme de Hunt et Szymanski

His speech was like a tangled chain ; nothing impair'd, but all disorder'd.
Who is next ?

Shakespeare — Le Songe d'une Nuit d'Été

▶ Soit E un ensemble fini muni d'une relation d'ordre partiel $\leqslant$. Une *chaîne* est une partie de E formée d'éléments deux à deux comparables ; une *antichaîne* est une partie de E formée d'éléments deux à deux incomparables. Une *décomposition en antichaînes* de E est une partition de E en antichaînes ; une telle décomposition est *minimale* si le nombre d'antichaînes qui la constituent est minimal.

Question 37 • Justifiez : une chaîne est de longueur maximale si et seulement si elle rencontre chaque antichaîne participant à une décomposition minimale.

▶ Un couple $(i,j) \in [\![0,m]\!] \times [\![0,n]\!]$ est un *accord* si $x_i = y_j$. Le *rang* d'un tel couple est $L(i,j)$; conventionnellement, $(0,0)$ est un accord (de rang nul). Un accord (i,j) est k-dominant s'il est de rang k et si tout autre accord (i',j') de rang k vérifie $i' > i$ et $j' \leqslant j$, ou $i' \leqslant i$ et $j' > j$. On note $p = \max(m,n)$.

▶ La complexité temporelle de l'algorithme de Hirschberg est proportionnelle au produit mn des longueurs des deux mots. Ceci est rédhibitoire avec des mots très longs. L'algorithme de Hunt et Szymanski a une complexité temporelle $\mathcal{O}\big((r+p)\lg p\big)$ (où r est le nombre d'accords entre les deux mots) tout en conservant une complexité spatiale linéaire. Cet algorithme a été employé par McIlroy en 1977 pour améliorer les performances de `diff`.

Question 38 • Montrez que, pour tout k compris entre 0 et $L(m,n)$ inclus, il existe au moins un accord k-dominant.

Question 39 • Avec $x = acbcab$ et $y = bccaacba$, dressez un tableau à sept lignes et neuf colonnes, dans lequel vous indiquerez le rang de chaque accord ; vous mettrez en évidence les accords dominants.

Question 40 • Montrez que la connaissance des accords dominants de rang k compris entre 1 et $L(m,n)$ inclus permet de déterminer un plus long sous-mot commun à x et y.

▶ Soient (i,j) et (i',j') deux accords ; on dira que (i,j) *précède* (i',j') si $i < i'$ et $j < j'$. La clôture réflexive de la relation binaire ainsi définie est une relation d'ordre sur l'ensemble E des accords.

Question 41 • Montrez que si l'on connaît une décomposition minimale de E en antichaînes vis-à-vis de cette relation d'ordre, on peut en déduire un plus long sous-mot commun à x et y.

▶ Pour $0 \leqslant i \leqslant m$ et $0 \leqslant \ell \leqslant n$, on note $k_{i,\ell}$ le plus petit indice j tel que $L(i,j) = \ell$ si toutefois un tel indice existe ; sinon, $k_{i,\ell} = n+1$.

Question 42 • Justifiez : si $k_{i,\ell} \leqslant n$, alors $k_{i,\ell}$ est la longueur du plus court préfixe de y ayant un sous-mot de longueur ℓ en commun avec $x[1..i]$.

Question 43 • En déduire (toujours sous l'hypothèse $k_{i,\ell} \leqslant n$) l'encadrement $k_{i,\ell-1} < k_{i+1,\ell} \leqslant k_{i,\ell}$, pour $0 \leqslant i < m$ et $1 \leqslant \ell \leqslant n$.

Question 44 • Comment pouvez-vous déduire un plus long sous-mot commun à x et y, en lisant le tableau des valeurs de $k_{i,\ell}$ associé à ces deux mots ?

Question 45 • Construisez le tableau des valeurs de $k_{i,\ell}$ associé aux mots $x = acbcab$ et $y = bccaacba$.

Question 46 ⋆⋆⋆ • Expliquez comment construire, pour un coût $\mathcal{O}(p \lg p)$, une structure de donnée fournissant, pour chaque valeur de i appartenant à $[\![1, m]\!]$, la liste $(j_1, j_2, \ldots, j_{q_i})$, classée en ordre décroissant, des indices j tels que (i, j) soit un accord.

Question 47 ⋆⋆⋆ • En déduire une mise en œuvre de l'algorithme de Hunt et Szymanski respectant les contraintes de complexité citées plus haut.

8 Questions bonus

Was this inserted to make interest good?

Shakespeare — Le Marchand de Venise

Question 48 • Soit s une permutation de $[\![1, n]\!]$. Montrez comment déterminer la plus longue sous-suite croissante extraite de la suite $\big(s(1), s(2), \ldots, s(n)\big)$ pour un coût quadratique par rapport à n.

Question 49 ⋆⋆⋆ • Montrez que, de toute permutation s de $[\![1, n]\!]$, on peut extraire une sous-suite monotone de longueur $\lceil \sqrt{n} \rceil$.

FIN