

Liechtenstein

La distance de Levenshtein entre deux chaînes de caractères a et b sur un alphabet Σ est définie par le nombre minimum d'opérations d'insertion, de suppression, ou de substitution d'une lettre pour passer de a à b .

1. Donnez un algorithme récursif pour calculer la distance de Levenshtein.
2. Montrez que la distance de Levenshtein est une distance, c'est-à-dire que :
 - $d(a, a) = 0$ et $d(a, b) = 0 \Rightarrow a = b$.
 - $d(a, b) = d(b, a)$.
 - $d(a, b) \leq d(a, c) + d(c, b)$.
3. Donnez un algorithme en $\mathcal{O}(|a| |b|)$ pour calculer la distance de Levenshtein. On pourra notamment trouver une relation de récurrence sur la distance de Levenshtein entre des mots et de leurs sous-mots.
4. Proposez des modifications de la distance de Levenshtein pour prendre en compte des typos comme “totu” ou “toout”.

Nombres tout pas beaux

On dit qu'un nombre entier positif est *tout pas beau* s'il a un facteur premier autre que 2, 3 ou 5. On dira qu'il est *tout beau* sinon.

1. Montrez que si k et l sont tout beaux, alors kl est tout beau.
2. Donnez un algorithme qui vérifie si un nombre est tout pas beau.
3. Donnez un algorithme qui renvoie le n -ème nombre tout beau.

Un nombre est *vraiment super beau* si tous ses facteurs premiers sont dans le tableau **primes**.

4. Adaptez votre algorithme pour renvoyer le n -ème nombre vraiment super beau.

Groupe Géométrique

On dit qu'une partie finie S d'un groupe (non nécessairement abélien) G engendre finiment G si et seulement si pour tout $g \in G$ il existe des $i_k \in \mathbb{Z}$ tels que $g = \prod_{s_k \in S} s_k^{i_k}$. On notera ceci $G = \langle S \rangle$

1. Montrez que $\mathbb{Z} = \langle 1 \rangle = \langle 2, 3 \rangle$.

On définit le graphe de Cayley de $G = \langle S \rangle$ noté $\Gamma(G, S)$ comme le graphe non dirigé, dont les sommets sont les éléments de G et il y a une arête entre g_1, g_2 si et seulement si $g_1 = sg_2$ pour un certain $s \in S$.

2. Dessinez $\Gamma(\mathbb{Z}, 1)$ et $\Gamma(\mathbb{Z}, \{2, 3\})$.

3. Dessinez $\Gamma(\mathbb{Z}^2, \{(0, 1), (1, 0)\})$.

On définit la distance des mots sur le groupe G relativement à S comme la distance sur $\Gamma(G, S)$.

4. Calculez la distance entre k, l sur $\mathbb{Z}$ relativement à 1. Calculez la distance sur $\mathbb{Z}^2$ relativement à $(0, 1), (1, 0)$.

5. Donnez un algorithme pour calculer la distance des mots sur G, S . Il devra être en $\mathcal{O}(d_{G,S}(a, b) |S|)$.

On dit que $f : X \rightarrow Y$ est bi-Lipschitz s'il existe $K \geq 1$ tel qu'on ait toujours :

$$\frac{1}{K} d_X(x_1, x_2) \leq d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$$

Si f est surjective, on dit que c'est une équivalence bi-Lipschitz.

6. Montrez le théorème suivant :

Théorème 0.1 Si G est finiment engendré par S et S' alors la fonction identité $(G, d_{G,S}) \rightarrow (G, d_{G,S'})$ est une équivalence bi-Lipschitz.

Solution: Groupe Géométrie

Voir https://www3.nd.edu/~gszekely/rtg/GTS/www3.nd.edu/_jquigle2/GSTS%20FA18/Week2.pdf