U.S.T.H.B - 2024-2025
Faculté d'Informatique
N. Bensaou & M.N Djouada

Master Informatique Visuelle
Algorithmique et Complexité (M1 - MIV)

Travail TP : Problème de la Plus Longue Sous-Séquence Commune (LCS)

1 Présentation du Problème

Énoncé du Problème : Étant données deux chaînes de caractères X de longueur m et Y de longueur n, la *plus longue sous-séquence commune (LCS)* est la plus longue séquence de caractères qui apparaît dans les deux chaînes dans le même ordre relatif mais pas nécessairement de manière contiguë. Par exemple, si $X =$ "$AGGTAB$" et $Y =$ "$GXTXAYB$", alors la LCS est "GTAB" avec une longueur de 4.

Objectifs :

- Implémenter plusieurs algorithmes pour calculer la LCS entre deux chaînes de caractères.
- Analyser et comparer les complexités en temps et en espace de chaque approche de manière empirique.
- Discuter des conditions pouvant optimiser ou simplifier le problème de LCS.

Importance : Le problème de la LCS est fondamental en informatique et possède de nombreuses applications dans la comparaison de données, la recherche de motifs et l'alignement de séquences.

2 Questions

- **Développement d'Algorithmes**
  1. Implémenter une solution récursive pour la LCS sans aucune optimisation. Analyser la complexité en temps et en espace de cette solution.
  2. Implémenter une solution en utilisant une matrice pour stocker les résultats intermédiaires. Analyser la solution, en notant la complexité spatiale.
  3. Peut-on optimiser la complexité spatiale de la solution précédente ? Analyser la complexité en temps et en espace de cette solution.
- **Analyse Empirique de la Complexité**
  1. **Comparaison de la Performance** :
     - Choisissez un ensemble de cas de test avec des longueurs de chaîne variables.
     - Enregistrez et analysez le temps d'exécution et l'utilisation de la mémoire de chaque approche pour chaque taille d'entrée.
     - Créez des graphes pour montrer la croissance empirique des performances en temps et en espace en fonction de la taille de l'entrée.
  2. **Analyse des Optimisations**
     - Identifier des conditions permettant de simplifier le calcul de la LCS. Par exemple, si une chaîne est beaucoup plus courte que l'autre ou si l'alphabet est limité (par exemple, chaînes binaires). discuter l'impact de ces facteurs sur la complexité.
     - Donner exemples empiriques de la réduction de la complexité.
  3. **Exploration d'applications du monde réel** :
     - Considérer un exemple à partir de données du monde réel (par exemple, la comparaison de séquences génétiques, le suivi des changements dans de grandes bases de code) et tester vos solutions.
     - Décriver comment la taille et la complexité des données du monde réel peuvent influencer le choix de l'algorithme de LCS.

3 Livrables

- **Code** : Remettre code de chaque solution et l'application.
- **Rapport** :Inclure un rapport (max 5 pages) qui couvre :
  - L'analyse de la complexité et les résultats empiriques (graphiques inclus). En spécifiant l'environnement d'expérimentation (Matériel et Logiciel).
  - La discussion sur les optimisations et les conditions qui réduisent la complexité pour l'application.

Ce travail doit être remis au plus tard Le Mardi 12/11/24 à 00:00. Ce travail se fait en ***trinôme ni plus ni moins***.