Conception d'algorithmes et applications (LI325)

Projet de programmation

1 Modalités de réalisation du projet

Le projet se fait en binôme ou seul. Un rapport contenant les réponses aux questions posées devra être remis avec le code source de vos programmes **e**t la sortie de vos programmes sur l'exemple donné en annexe. Il devra exposer de manière claire et concise (5 pages maximum) les algorithmes choisis et les difficultés rencontrées. Il ne s'agit pas de commenter une seconde fois le code ni d'expliquer chaque fonction en détail (c'est le rôle des commentaires dans le code) mais d'en présenter une synthèse. Le rapport (au format PDF) ainsi que le source et la sortie de vos programmes (au format TXT) devra être fourni par courriel à l'adresse de votre chargé de TD au plus tard le **vendredi 26 avril à 12h**.

Les points suivants devront faire l'objet d'une attention particulière lors de l'écriture des programmes : sa correction, sa lisibilité et sa clarté.

Vous avez le choix quant au langage de programmation à utiliser.

Rappel :

Groupe 1 : `antoine.genitrini@lip6.fr`
Groupe 2 : `alice.albano@lip6.fr`

2 Impression équilibrée

Le problème est l'impression équilibrée d'un paragraphe sur une imprimante. Le texte en entrée est une suite de n mots de longueur $w_1, \ldots, w_n$ (la longueur d'un mot est le nombre de caractères de ce mot). On désire imprimer ce paragraphe de manière équilibrée sur des lignes ne pouvant contenir qu'un maximum de W caractères chacune. Le critère d'équilibre est le suivant.

> Si une ligne donnée contient les mots de i à j inclus et que nous laissons un caractère d'espacement entre chaque mot, le nombre de caractères d'espacement supplémentaires à la fin de la ligne est $W - j + i - \sum_{k=i}^{j} w_k$. L'objectif est de minimiser la somme, sur toutes les lignes *hormis la dernière*, des cubes des nombres de caractères d'espacement présents à la fin de chaque ligne.

Par exemple, pour équilibrer la phrase suivante : *"Je fais le projet de LI325 tout seul"* avec $W = 10$, une solution est :

```
Je_fais___
le_projet_
de_LI325__
tout_seul_
```

Le déséquilibre de cet agencement est $3^3 + 1^3 + 2^3 = 36$ (on ne compte pas la dernière ligne). Les caractères de ponctuation sont à traiter comme des caractères normaux : il ne peuvent être séparés des mots adjacents que s'il y a un espace entre eux. Par exemple, `c'est-à-dire` constitue un seul mot inséparable.

1. Programmer l'algorithme glouton consistant à remplir les lignes une à une en mettant à chaque fois le maximum de mots possibles sur la ligne en cours. Est-ce que cet algorithme fournit l'optimum ?
2. Donner un algorithme de type programmation dynamique résolvant le problème. Analyser sa complexité en temps et en espace.
3. Supposons que pour la fonction de coût à minimiser, on ait pris simplement la somme des nombres de caractères d'espacement présents à la fin de chaque ligne. Est-ce que l'on peut faire mieux en complexité que pour la question 2 ?
4. Modifier l'algorithme de la question 2 afin de prendre comme fonction de coût la somme puis la somme des carrés des nombres de caractères d'espacement présents à la fin de chaque ligne. Comparer visuellement le résultat obtenu sur le texte donné en annexe.

3 Annexe

Vous testerez votre programme au moins sur l'exemple suivant, avec $W = 20$ et $W = 50$. Annotez à la sortie le nombre de lignes de votre solution et le déséquilibre de votre agencement.

```
Inventée par le professeur Richard Bellman,
la programmation dynamique permet de
résoudre au moyen d'un ordinateur tout problème
d'optimisation
dont la fonction objectif se décrit
comme la somme de fonctions monotones
non-décroissantes des ressources.
Concrètement, cela  signifie que l'on va pouvoir
déduire la
solution optimale d'un problème à partir d'une solution
optimale d'un sous problème.
On appelle algorithme glouton un algorithme qui suit le
principe de faire, étape par étape, un choix optimum local,
dans l'espoir d'obtenir un résultat optimum global.
Dans les cas où l'algorithme ne fournit pas
systématiquement la solution optimale, il est appelé
une heuristique gloutonne.
```