TP AOD

Ensimag 2A

Date de rendu Teide : mercredi 12 Novembre 2014

En typographie, un paragraphe est *justifié* si les extrémités des lignes sont alignées à gauche et à droite en forçant la dernière ligne à gauche. Il s'agit donc de trouver un *découpage* du paragraphe en lignes, c'est à dire après quels caractères (on dit *signes* en typographie) mettre un retour à la ligne.

Un texte est une séquence de *caractères imprimables* ou de *séparateurs*. Les caractères imprimables sont les caractères alpha-numériques, les symboles de ponctuation et les symboles graphiques (\$, %, ...). Les séparateurs sont l'espace (ES), le retour à la ligne (RC), la tabulation (TAB).

On appelle *mot* une séquence de caractères imprimables contigus. Un paragraphe est une séquence de mots séparés par des séparateurs et qui se termine par deux caractères retour à la ligne (éventuellement séparés par des espaces).

On cherche à imprimer un paragraphe justifié, en s'interdisant de couper un mot en deux parties par un retour à la ligne. Pour cela, on décide après quels mots est inséré un retour à la ligne. Le paragraphe est ainsi découpé en lignes, chacune contenant des mots.

La longueur de l'affichage d'un mot m est mesurée en *unité d'affichage* (en pixels, ou en points par exemple); elle est notée $L(m)$ et dépend de la police utilisée. Entre deux mots consécutifs sur une même ligne, il est nécessaire d'insérer un espace minimal de séparation de longueur μ unités d'affichage, qui dépend aussi de la police. Soit $[m_0, \ldots, m_n]$ un paragraphe avec $n+1$ mots. Supposons qu'une ligne affiche la séquence de mots consécutifs $[m_i, \ldots, m_k]$ (m_i est le premier mot de la ligne, et m_k le dernier mot). L'affiche de cette ligne requiert une longueur minimale $\Delta(i,k)$ unités définie par

$$\Delta(i,k) = L(m_k) + \sum_{j=i}^{k-1} (L(m_j) + \mu).$$

A l'exception de la dernière ligne du paragraphe, toutes les lignes sont appelées *internes* et ont la même longueur égale à M unités. Soit D un découpage sur l lignes, de 1 à l. Soit une ligne interne j ($1 \leq j < l$) dont le premier mot est m_i et le dernier m_k. On a nécessairement $L(i,k) \leq M$ et pour aligner le paragraphe on ajoute une taille d'espace $e_j = E(i,k) = M - L(i,k)$ unités; cette taille d'espace est répartie uniformément[1] entre les mots de la ligne.

On définit le coût de ce découpage par $C(D) = \sum_{j=1}^{l-1} N(e_j)$, où $N(x)$ est une fonction de pénalité imposée (croissante, positive et entière); par exemple $N(x) = x^3$ (d'autres fonctions pourront être utilisées lors de l'évaluation). On *ne compte pas* dans ce coût le nombre d'espaces laissés sur la dernière ligne, celle-ci pouvant être incomplète sans gêner le lecteur.

Pour obtenir un affichage agréable, on cherche un découpage du paragraphe P de coût minimal :

$$C^\star(P) = \min_{D\text{: découpage de } P} C(D).$$

La justification d'un texte F consiste à aligner optimalement chacun de ses paragraphes; le coût d'alignement du texte est la somme des coûts des découpages de chacun de ses paragraphes :

$$C^\star(F) = \sum_{P\text{: paragraphe de } F} C^\star(P).$$

[1] Par exemple, on met une longueur $(e_j \div (k-i)) \cdot \mu$ entre les mots de la ligne; et on augmente (pseudo-aléatoirement) ($e_j \mod (k-i)$) de ces espaces d'une longueur μ; par exemple cycliquement avec décalage (on parle de *stride*). Exemple : si la ligne contient 5 mots et $e_j = 6$ et avec un stride de 2 à partir du 2ème mot m_{i+1}, on ajoutera un espace de longueur $(6 \div 4)\mu = \mu$ entre : m_i et m_{i+1}; m_{i+2} et m_{i+3}. Et 2μ entre : m_{i+1} et m_{i+2}; m_{i+3} et m_{i+4}.

On pose

$$\phi(i) = \begin{cases} 0 \text{ si } E(i,n) \geq 0, \\ \min_{k \geq i : E(i,k) \geq 0} \left(\phi(k+1) + N(E(i,k))\right) \text{ sinon.} \end{cases}.$$

Travail à rendre

Il est demandé de construire un algorithme qui justifie optimalement un fichier au format UTF-8, de l'analyser et de le programmer. Le choix du langage est laissé libre. Néanmoins, une version C du fichier `altex.c` est fournie contenant un `main` qui analyse les paramètres en entrée du programme. De plus, un fichier `altex-utilitaire.c` en langage C fournit des primitives utiles pour le problème.
L'archive fournie contient:

- le questionnaire à compléter (en latex ou PDF);
- un fichier de commande `test-altex` qui ne doit pas être modifié (ce programme appelle le programme demandé `altex`);
- un programme source `altex.c` contenant la fonction `altex` à compléter;
- un makefile générant l'exécutable;
- les fichiers `altex-utilitaires.h` et `altex-utilitaires.c` qui contiennent des primitives permettant de : lire un fichier mot par mot ; calculer la longueur de l'affichage d'une chaîne de caractères ; calculer N (la fonction de pénalité). afficher (en TXT ou en PDF) une séquence de mots avec une longueur d'espace donnée sur une ligne interne (la longueur est alors équirépartie entre les mots) ou sur la dernière ligne du paragraphe. Il est fortement recommandé d'utiliser ces primitives car elles permettent de simplifier l'écriture de votre programme. Ces deux fichiers ne doivent pas être modifiés; lors de l'évaluation automatique, ils seront remplacés par d'autres fichiers respectant les mêmes spécifications ;
- des fichiers texte exemples à aligner.

On demande:

- de remplir le questionnaire (1 page au maximum) en répondant aux questions;
- de programmer en langage C la fonction:
  `long altex(FILE* in, size_t len, struct stream *outformat, unsigned long M, unsigned N)`
  qui retourne le coût d'alignement du texte sur le fichier d'entrée `in` qui est écrit aligné sur le fichier de sortie `out`. Les paramètres sont:
  - `in` = le fichier en entrée (précondition: ouvert en lecture), par exemple `stdin`;
  - `len` = la taille du fichier en entrée;
  - `outformat` = une structure spécial permettant l'écriture dans un fichier au format TXT ou PDF grâce aux fonctions fournies par `altex-utilitaires.h`;
  - M = la longueur des lignes (par exemple $M = 80$) ;
  - N = la valeur de l'exposant ν (par exemple $\nu = 3$).
- le source du programme doit contenir au début une partie commentaire qui décrit : brièvement la méthode utilisée; les choix d'implantation suivis. En particulier, on expliquera combien de fois sont calculées les tailles de chacun des mots; et combien de fois les données sont parcourues en mémoire.

Important. Le programme test-altex sera utilisé pour évaluer automatiquement le résultat du programme rendu sur une batterie de fichiers test avec des fonctions de pénalités spécifiques. Il est impératif de ne pas le modifier (en cas d'échec ou de non conformité à un test, la note du test sera 0).

Barème

- Programme (source + résultats des tests automatiques) : 12 points
- Questionnaire 8 points