Licence d'informatique
Module de C/Unix

Projet C

Mots croiss

Jean-Luc LEVAIRE

Dcembre 2002

Le but de ce projet est de placer un ensemble de mots sur une grille de taille quelconque. Les mots peuvent tre placs soit horizontalement, et ils se lisent alors de gauche droite, soit verticalement, et ils se lisent de haut en bas. Deux mots peuvent s'intersecter (un mot vertical, un mot horizontal) aux positions o ils possdent des lettres communes. Un mot peut intersecter plusieurs mots. Tout mot plac sur la grille doit tre entour de cases vides, hormis pour les cases o il intersecte un autre mot, et ses extrmits o les cases en diagonale peuvent tre occupes par un autre mot (voir Fig. 1). Tout le problme est de trouver le meilleur placement rpondant un certain critre : taille minimale de la grille englobante, plus grand nombre d'intersections entre mots... . . .

Lecture de l'ensemble de mots

L'ensemble des mots placer est contenu dans un fichier texte, dont le nom est pass en paramtre du programme. Le fichier contient un mot par ligne. Les fonctions `fopen` et `fgets` permettent d'ouvrir et de lire ligne par ligne un fichier. Les mots sont lus un par un dans un buffer, allous dynamiquement et insrs dans une liste chane. Voici un exemple de programme utilisant ces deux fonctions et imprimant le contenu d'un fichier ligne par ligne.

```
#include <stdio.h>
#define MAX_LENGTH 80

int main(int argc, char *argv[]) {
  FILE *file;
  char buffer[MAX_LENGTH];
  /* Ouverture du fichier en lecture (r pour read */
  /* argv[1] (1er argument) est le nom du fichier   */
  if ((file = fopen(argv[1], "r")) == NULL) {
    fprintf(stderr, "%s: Error opening %s: ", argv[0], argv[1]);
    perror("");
    exit(2);
  }
```

				a	i	r	b	u	s			
				v								
	h	e	l	i	c	o	p	t	e	r	e	
				o								
b	o	e	i	n	g							

FIG. 1 – Un exemple de placement pour 4 mots. Les cases en gris sont celles qui ne peuvent tre occupes en raison de la prsence du mot hlicoptre.

```
  /* Lecture d'une ligne */
  while (fgets(buffer, MAX_LENGTH, file))
    printf("%s", buffer);
}
```

Recherche d'un meilleur placement

La solution triviale pour trouver un meilleur placement est de les parcourir dans leur totalit. Pour cela, on utilise l'algorithme suivant :

1. Placer le premier mot de la liste la position (0, 0) horizontalement. La grille G initiale des mots placs est donc rduite ce mot. Construire une seconde liste L des mots non encore placs ;
2. Pour chaque mot M de la liste L des mots non encore placs :
   – supprimer M de L ;
   – Si L est vide, pour chaque placement possible de M dans G, valuer le placement obtenu par rapport au critre, et le stocker si il est meilleur que celui actuellemnt obtenu. Si aucun placement de M dans G n'est possible, on n'a pas de solution ;
   – Si L n'est pas vide, on recommence de manire rcursive l'tape b pour chaque placement possible de M dans G. Si aucun placement n'est possible, on n'a pas de solution. Avant chaque appel rcursif, on modifie G en fonction du placement de M dans G ;
   – rinsrer M dans L ;

Le fait qu'un placement soit possible dpend des conditions numres dans la section prcdente et des mots dj places (i.e. la grille courante). Ici on cherchera les intersections du mot avec ceux dj placs, et pour chaque intersection, on vrifiera si elle est possible par rapport aux mots dj placs.

Quant au critre d'valuation d'un placement, on pourra choisir la taille minimale de la grille englobante. L'option `-i` du programme permet de changer le critre en le plus grand nombre d'intersections de mots.

Structures de donnes proposes

En plus de la liste des mots, l'algorithme prcdent utilise une liste de mots non placs et une grille de mots placs. Nous proposons de reprsenter une grille de mots placs par un tableau de structures reprsentant chacune un mot plac, et contenant :

– la position (x, y) du mot (i.e. de sa premire lettre) ;
– le sens de placement du mot : horizontal ou vertical ;
– la longueur du mot (pour acclerer la recherche des placements possibles) ;
– un pointeur sur la chane correspondant au mot ;

Un tableau est allou dynamiquement au dbut du programme, avec une longueur gale au nombre de mots lus dans le fichier. Ce tableau reprsentera la grille courante des mots placs. Une seconde grille est aussi alloue et contiendra le meilleur placement courant. L'ensemble de ces structures est reprsente Fig. 2.

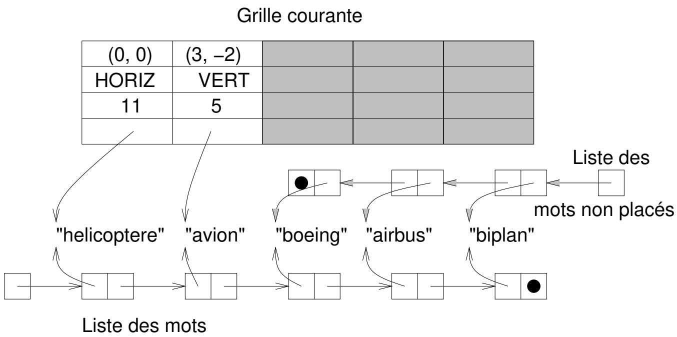

FIG. 2 – Les structures de donnes proposes.