TP 2 : tableaux et polynomes

Programmation en C (LC4)

Semaine du 4 février 2008

1 Polynomes

On représente un polynome par un tableau de `double` contenant ses coefficients (par exemple le coefficient de degré i dans la case d'indice i).

La libc fournit plein de fonctions pour traiter les nombres de types `double`. Pour les utiliser, il faut mettre un `#include<math.h>` au début de votre fichier `.c`, et ajouter un `-lm` sur la ligne de commande de compilation (par exemple `gcc -lm truc.c`). En particulier, la fonction `pow` qui calcule une puissance, et la fonction `floor` qui calcule la partie entière.

Exercice 1 Écrire une fonction `double evalue(double P[],int d,double x)` qui évalue le polynome `P` de degré d sur le nombre `x`.

Exercice 2 Écrire une fonction `double racine(double P[],int d, double a, double b, double precision)` qui calcule par dichotomie une valeur approchée à `precision` près d'une racine du polynome `P` de degré `n` situé entre `a` et `b`. On suppose que `P` n'a pas le même signe en `a` et en `b`.

Exercice 3 Écrire une fonction `void decalage(double P[],int d, double x)` qui transforme `P` en `P(X+x)`.

Exercice 4 Écrire une fonction `void trace(double P[],int d,double precision,double x,double y, int largeur, int hauteur,int image[])` qui dessine dans le tableau `image` le graphe du polynome `P` de degré `n`. Le coin inférieur gauche de l'image est en (x, y), l'image fait `largeur` pixels de large, `hauteur` pixels de haut, et la distance horizontale ou verticale entre deux pixels adjacents est de `precision`. Si vous affichez un point par abcisse, la courbe sera discontinue quand la dérivée est trop grande. Essayez d'y remédier.

Quelques rappels du ternier TP pour la manipulation des images :

On représente une image par un tableau d'entiers, chaque entier représentant la couleur d'un pixel. Ici, une couleur sera juste un entier entre 0 (noir) et 255 (blanc), représentant un niveau de gris. Les pixels sont donnés dans l'ordre suivant :
– le premier pixel est le coin supérieur gauche
– suivent les pixels de sa ligne, de gauche à droite
– suit la deuxième ligne, toujours de gauche à droite
– et ainsi de suite jusqu'en bas
Le pixel de coordonnées (x, y) s'obtiendra donc par `image[x+width*y]`.

Vous pouvez récupérer les fichiers `ppm.c`, `ppm.h` à l'adresse :

`http://cedric.cnam.fr/~lambe_a1/C.html`

Le fichier `ppm.c` fournit les deux fonctions suivantes :

void charge_image_ppm (**const char**∗ filename, **int** width, **int** height, **int** image[]);
void enregistre_image_ppm (**const char**∗ filename, **int** width, **int** height, **const int** image[])

qui permettent de charger depuis ou enregistrer dans un fichier une image.

Il faudra avoir au préalable compilé `ppm.c` avec la commande `gcc -c ppm.c` (à faire une seule fois), puis compiler votre `.c` (mettons que vous l'avez appelé `image.c`) à vous avec `gcc ppm.o image.c` (en fait, `gcc -lm ppm.o image.c`).

2 Un algorithme de tri

Exercice 5 Écrire une fonction `int separe(unsigned int i,int j,unsigned int t[],int x)`, qui réordonne le tableau `t` entre les indices `i` et `j`, de manière à regrouper en premier tous les éléments dont le $\texttt{x}^{\text{ème}}$ bit vaut 0, et donc en second tous ceux dont le $\texttt{x}^{\text{ème}}$ bit vaut 1. La fonction doit renvoyer en résultat l'indice du premier élément dont le $\texttt{x}^{\text{ème}}$ bit est à 1.

On suppose que le type `unsigned int` fait 32 bits. (C'est le cas sur les PCs.)

À l'aide de la fonction `separe`, il est possible de trier un tableau :
- on commence par séparer sur le trente deuxième bit
- les éléments du premier paquet sont inférieurs à ceux du second, puisque ceux du premier sont inférieurs à 2^{31}, tandis que ceux du second sont supérieurs ou égaux à 2^{31}
- il suffit donc de trier chacun des deux paquets indépendement
- pour ce faire, on procède récursivement à l'intérieur de chaque paquet, en séparant cette fois sur le trente et unième bit
- et ainsi de suite jusqu'au dernier bit

Exercice 6 Écrire une fonction prenant en argument un tableau et le triant par la méthode décrite ci-dessus.