Suppression de points

Bruno Malenfant

15 mai 2012

- Peut être fait seul(e) ou en équipe de deux (2).
- Date de remise : 13 juin 2014 avant minuit.

1 Introduction

Pour le premier tp vous allez construire un logiciel qui réduit le nombre de points décrivant une courbe. Votre logiciel va recevoir une liste de points (coordonnées cartésiennes) et enlever les points qui vont causer une perte d'information minimum. Voici une description des entrées, du traitement et des sorties de ce logiciel.

2 Description

2.1 Entrées

Les entrées du logiciel sont données sur la ligne de commande lors du lancement et au clavier durant l'exécution du logiciel. Sur la ligne de commande, le logiciel va recevoir une valeur entière entre 1 et 100 indiquant le pourcentage (p) des points à conserver. Les arguments de la ligne de commande sont placés par l'OS dans les arguments de la fonction `int main( char *, int )`. Si l'utilisateur lance le logiciel sans valeur sur la ligne de commande ou que la valeur donnée n'est pas un entier entre 1 et 100 alors le logiciel termine avec un message d'erreur sur le canal d'erreur. Pour afficher un message sur le canal d'erreur, vous allez utiliser la commande suivante :

```
fprintf( stderr, "un message d'erreur" );
```

Quand un logiciel termine avec une erreur sur unix/linux, il est demandé que le logiciel retourne une valeur négative. Nous pouvons terminer un logiciel en tout temps avec la commande `exit`, par exemple :

- exit(-1);

Ensuite, les coordonnées des points de la courbe sont lues au clavier. L'utilisateur va entrer une première valeur entière indiquant le nombre de points (n) qu'il y aura en entrées. Cette valeur sera suivie de la touche 'enter' et elle doit être plus grande que 1. Toutes entrées non conformes doivent avoir un message d'erreur et le logiciel doit se terminer.

L'utilisateur entre les données à raison de 1 point par ligne. Un point est représenté par ses coordonnées en x_i et en y_i. Les deux valeurs (x_i, y_i) sont séparées par un espace. Vous devez utiliser des doubles pour représenter ces valeurs. Les valeurs en x doivent être données en ordre croissant. Voici un exemple d'entrées :

```
>./a.out 90
10
-4.5 2.0
-4 3.0
-3.5 4.0
-3 2.0
-2.5 1
-2 -40.4
-1 12
3 -4.0423434
6 2.098
15 1.49
```

Remarquez l'absence de message de la part du logiciel. Votre logiciel doit être silencieux, seuls les messages d'erreur servent d'interaction.

2.2 Analyse

2.2.1 Calcul du nombre de points à supprimer

Votre logiciel doit choisir les points à enlever. Premièrement, il faut déterminer le nombre de points à conserver (n_c). Il suffit de multiplier n par $\frac{p}{100}$ et d'arrondir à l'unité la plus proche. Vous devez toujours conserver un minimum de 2 points. Si le résultat est plus petit que 2 alors prenez 2 comme résultat. Le nombre de points à supprimer sera donc $n_s = n - n_c$.

2.2.2 Sélection du point le moins important

Pour choisir le point le moins important, il suffit de suivre les étapes suivantes :

1. Calculer les pentes des segments pour chaque segment.

$$d_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

2. Calculer les angles équivalents.

$$a_i = \arctan(d_i)$$

3. Calculer les différences d'angle pour chaque point.

$$\Delta_i = |a_{i+1} - a_i|$$

4. Trouver le Δ_i minimum, c'est le point à supprimer.

Vous devez répéter ces étapes jusqu'à ce que le nombre de points voulu aient été supprimés.

2.2.3 Exemple de calcul

i	0	1	2	3	4
x_i	-2	-1	0	1	2
y_i	-0.5	2.2	1.6	1.8	1.0
d_i	-	2.7	-0.6	0.2	-0.8
a_i	-	1.216	-0.540	0.197	-0.675
Δ_i	-	1.756	0.737	0.872	-

Il faudra enlever le point $i = 2$ qui a le plus petit Δ.

2.3 Sorties

Pour les sorties, votre logiciel doit premièrement afficher le nouveau nombre de points n_c sur une ligne. Ensuite, les points conservés sont affichés à raison de 1 par ligne. Pour chaque point, ces coordonnée en x et en y sont affichées séparées par un espace. Utilisez le format suivant pour afficher les doubles :

```
printf( "%8f", votre_variable )
```

3 Conception

Chacune de vos routines doit avoir des commentaires. Ces commentaires doivent avoir l'information suivante :

- Rôle de la routine : ce qu'elle fait, ce qu'elle retourne.
- Argument : Description des arguments, leurs rôles dans la routine, les valeurs permises, sont-ils des entrées ou des sorties ? ...
- Les erreurs possibles.

Votre programme doit être dans un seul fichier.

4 Remise

N'oubliez pas d'écrire vos noms dans le code. Remettez votre programme en utilisant Moodle. Votre programme doit être dans un seul fichier. Vous n'avez le droit d'utiliser que les librairies suivantes pour votre logiciel : stdio.h, stdlib.h, string.h, math.h, stdbool.h.

5 Évaluation

- Fonctionnalité (12 pts).
  - Votre programme doit compiler avec le compilateur 'gcc' sur la machine rayon. Si le programme ne compile pas, aucun point n'est donné pour la fonctionnalité.
- Structure (4 pts).
  - Le respect des critères pour les commentaires.
  - Bonne utilisation des structures de contrôle : if, while, ...
  - Bonne modularité.
- Lisibilité (4 pts).
  - Bonne indentation.
  - Identificateur significatif.
- Autres critères :
  - Pas de 'goto' et 'continue'.
  - Un seul 'return' par routine, à la fin de celle-ci.
  - Pas de variables globales.
  - Pas de ligne contenant des commentaires et du code.