

Répartition des tâches :

Charles - La gestion des graphiques (Gnuplot), le script shell (message "Help"), Le ReadMe et aide sur le C.

Hicham - La préparation du traitement T et S en C (AVL)

Abhishek - la majeure partie du script shell, l'organisation des fichiers et la combinaison de chacun de nos codes.

Le Déroulement :

Les deux premières semaines de décembre, tout le monde démarre sur le script shell, nous faisons des recherches chacun de notre côté.

Charles recherche les histogrammes en Gnuplot, le script shell de départ (les messages d'aide , comment mettre le temps d'exécution etc).

Hicham fait part à l'équipe qu'il a des difficultés avec le script shell et propose de commencer le AVL pour les traitements T et S.

Abhishek commence le premier main.sh avec des vérifications et le combine avec le main.sh de Charles. Il va continuer de gérer les différents traitements d1/2 et L avec la fonction sort .

A partir du 15 Décembre, Charles partage les premiers histogrammes gnuplot fonctionnel sur le github. Hicham continue de travailler sur les AVL, Il partage la première version de son AVL cette semaine. Abhishek a un souci avec son émulateur et doit trouver une nouvelle solution pour coder durant cette semaine. Il y arrive en milieu de semaine et reprend son main.sh.

A partir du 17 Décembre, on ne partage plus rien jusqu'au 26 car nous avons notre semaine d'examens du 22 au 24 à préparer. A partir de ce moment , Abhishek a pu compiler mes gnuplots avec son main.sh et va mettre sur le github son main.sh avec les traitements d1/2 et L fonctionnel sur le github. Il va aussi réorganiser les fichiers dans des dossiers comme demandé(data,prog etc). Ensuite, Hicham va envoyer sa version du AVL et nous allons essayer de la faire marcher avec le script shell.

Nous nous sommes souvent appelés pour discuter du C et du script shell ensemble, pour regarder ce qu'on pouvait simplifier ou mettre, ce qui explique les nombreux changements de code C. Pour mettre les modifications sur github, Abhishek étant le plus à l'aise, va recevoir nos fichiers et va les partager après les avoir reliés avec ses travaux, ce qui explique pourquoi la plupart du temps, les changements sur le site ont été faits par lui. Le C fonctionne mais on avait des problèmes dans les valeurs. L'option -T ne fonctionnait pas correctement (pas les mêmes valeurs etc.). Abhishek a essayé de refaire l'option -t de son

côté et maintenant le problème est résolu. Dorénavant, le projet fonctionne mais les traitements T et S sont longs.

Nous sommes le 02/02, jour du rendu, Charles termine les derniers commentaires en anglais et vérifie que tout fonctionne. Il revoit le pdf

Ce qui fonctionne pas ou mal :

- Le temps des traitements -d1, -d2 et -l est inconsistent. -d1 peut aller de 7 à 12 secondes, -d2 de 3 à 8 secondes, et -l de 7 à 9 secondes. On a testé la plupart de nos traitements sur une machine virtuelle. Celle-ci étant lente, on pense donc que cela dépendra de la performance de l'ordinateur.
- L'exécution des traitement -t et -s qui prend entre 30 à 60 secondes (sur oracle virtual box , ca prend plusieurs minutes). Cela est dû au fait que lire le fichier prend beaucoup de temps.

Démo

Traitement -d1

Pour le traitement -d1, on obtient les mêmes valeurs que les fichiers partagés par notre professeur.

```
demo > temp > ≡ data_d1.dat
1  Angela FERNANDE;212
2  Andrea ZIMMERMAN;205
3  Vera MOREN;194
4  Jose HUBE;193
5  Giovanni JENSE;182
6  Michel FERRAR;181
7  Lyubov TÔT;175
8  Thomas GUTIERRE;174
9  Richard IVANO;172
10 Mariya SANCHE;166
```

Traitement -d2

Pour le traitement -d2, on obtient pas les mêmes valeurs que les fichiers partagés par notre professeur, cependant on obtient la même liste de conducteurs.

```
demo > temp > ≡ data_d2.dat
1  Andrea ZIMMERMAN;146260,000000
2  Angela FERNANDE;141251,000000
3  Vera MOREN;139635,000000
4  Jose HUBE;137729,000000
5  Giovanni JENSE;127981,000000
6  Lyubov TÓT;126019,000000
7  Thomas GUTIERRE;123433,000000
8  Sandra GOME;119352,000000
9  Michel FERRAR;118373,000000
10 Richard IVANO;116777,000000
```

Traitement -l

Même cas avec -d2

```
demo > temp > ≡ data_l.dat
1  295579;2477,000000
2  256594;2307,000000
3  252014;2328,000000
4  227572;2487,000000
5  137171;2812,000000
6  91908;2307,000000
7  66827;2336,000000
8  31409;2776,000000
9  23484;2297,000000
10 9858;2419,000000
```

Traitement -s

Pour le traitement -s, on obtient les même valeurs que le professeur (le fichier étant long, on ne peut pas mettre tous les valeurs sur l'image)

```
demo > temp > ≡ data_s.dat
1  1;172705;3.865000;108.875381;964.148987;960.283997
2  2;31409;2.242000;107.314972;942.931030;940.689026
3  3;200110;9.865000;101.584351;877.786011;867.921021
4  4;32430;3.149000;66.825867;826.698975;823.549988
5  5;30644;90.222000;496.277985;902.333984;812.112000
6  6;59689;1.918000;73.772659;806.265991;804.347961
7  7;286095;56.715000;455.787506;854.859985;798.144958
8  8;12167;32.462002;431.335480;830.208984;797.747009
9  9;180463;2.923000;67.124397;800.070984;797.148010
10 10;48290;47.926998;445.732513;843.538025;795.611023
```

Traitement -t

Pour le traitement -t, on obtient les même valeurs que le professeur, excepté pour Marseille qui manque une route et STE Colombe qui est tout en bas de la liste

```
demo > temp > ≡ data_t.dat
1  ANDILLY;1900;417
2  BEAULIEU;1998;25
3  LE PIN;2103;27
4  MARSEILLE;3400;1011
5  MONS;1942;74
6  SALLES;2151;382
7  ST LEGER;1820;44
8  ST MICHEL;1909;168
9  ST SAUVEUR;3043;350
10 STE COLOMBE;2789;113
```