

PDF CY-TRUCK



Eva ANSERMIN
2023-2024

Matthieu BRAULT
Ostrean BONNET
Wissam IZEM



Pour commencer notre projet, nous avons décidé de venir en td (matthieu et ostrean, wissam ne pouvant pas venir) afin d'accumuler aucun retard. Lors de ces séances de td, nous avons réussi à faire les fonctions d1, d2 et l. Nous nous sommes mutuellement aidés avec certains groupes afin que l'exécution du code soit la plus rapide possible. Nous avons cependant rencontré des difficultés assez conséquentes par rapport à l'utilisation de la fonction 'awk', chat.gpt nous a bien aidé à la comprendre et à savoir l'utiliser. Donc nous avons réussi à avoir nos trois fonctions (d1, d2, l) juste avant les vacances de Noël.

Afin de pouvoir travailler efficacement pendant les vacances, nous nous sommes répartis le travail de la manière suivante: -Matthieu devait s'occuper du traitement_s et du traitement_t en c -Wissam devait s'occuper de la partie shell (et plus particulièrement de la robustesse de notre code) -et enfin Ostrean devait s'occuper des graphiques. Cette répartition organisée du travail nous a permis d'avancer correctement pendant les vacances, même si c'était plus dur d'avancer car il y avait peu/pas d'aides extérieures possibles (camarade et ou prof).

Dans ce paragraphe je vais expliquer les nombreuses difficultés rencontrées concernant les graphiques. J'avais décidé de commencer par le graphique horizontal de d2 (car on avait déjà la fonction avant les vacances). Afin d'apprendre à faire un graphique je me suis renseigné sur chat.gpt, des forums, et le site de gnuplot. Cependant ce fut très compliqué car au début je ne connaissais pas l'existence de la fonction convert (qui permet de faire une rotation à une image), ducoup j'avais consulté beaucoup de forums anglophone (car malheureusement il n'y avait pas de forums français sur ça) pour réussir mes graphs et sur aucun d'entre eux la fonction convert n'était pas mentionnée (<https://stackoverflow.com/questions/20408106/gnuplot-plotting-histograms-on-y-axis>). J'ai donc passé mes vacances à essayer d'y arriver en vain, jusqu'au moment du premier td de 2024 où des camarades de classe m'ont appris l'existence de cette fonction. Cela a été l'élément déclencheur qui m'a permis d'avancer considérablement, en une soirée j'avais réussi à faire d1 et d2. Puis la soirée d'après j'ai fait le graph de l assez facilement. Ensuite je me suis attaqué au graph de s, la aussi ce fut assez compliqué car je n'arrivais pas remplir l'espace entre les courbes min et max (même en créant une seule fonction), ducoup j'ai réussi à trouver une alternative en remplissant toute la partie en dessous de la courbe max (avec filledcurves above) puis en remplissant toute la partie en dessous de la courbe min en blanc (toujours avec la même fonction) afin de se rapprocher le plus possible du graphique donnée. Le seul problème de cette méthode c'est que ça remplit tout ce qu'il y a en dessous en blanc même les petites barres sur l'axe des x. Quant au graphique de t, il n'y avait pas de difficultés nouvelles hormis le fait de superposer les boxes. J'ai eu un petit peu de mal à y arriver jusqu'au moment où j'ai trouvé cette fonction sur youtube (set style fill transparent solid 0.5) qui m'a permis de grandement avancer et de réussir mon graphique.

-Dans ce paragraphe je vais vous expliquer les difficultés rencontrées avec le script shell. Il faut savoir que je n'avais pas été présent lorsque qu'on a commencé à l'étudier. J'ai donc rencontré des gros problèmes à comprendre et coder en shell. J'ai passé une grande partie de mes vacances à essayer de rattraper mon retard sur le sujet. Enfin j'ai réussi pendant la semaine en distanciel à faire des choses relativement simples tel que dire si l'argument passé en paramètre était valide ou non et à quoi il correspondait. L'utilisation du cm et de chat.gpt m'a permis de comprendre beaucoup plus vite. Cependant je n'arrivais toujours pas à faire le reste (créer les fichiers, -h, et tout le reste du cahier des charges). Au retour des vacances, j'ai fait part de mes difficultés au groupe, et matthieu (étant le plus fort en shell) s'est chargé de ce qu'il me restait à faire et a essayé de m'expliquer en même temps. Pour le makefile j'ai pris celui de l'année dernière et j'ai fait juste quelques modifications pour que ça marche.

Dans ce dernier paragraphe, je vais vous expliquer comment j'ai réalisé les traitements T et S. Tout d'abord, je pensais que nous n'avions pas du tout le droit d'utiliser le script Shell pour nous aider. J'ai donc commencé directement en langage C, en lisant les lignes du fichier data.csv et en décodant les valeurs grâce aux commandes apprises sur le site KooR.fr. Cela m'a permis de commencer à implémenter mes AVL, en reprenant les fonctions du cours. J'ai créé un AVL qui trie les villes par ordre alphabétique.

Par la suite, j'ai dû prendre en compte les doublons de lignes du fichier, ce que j'ai géré dans la fonction d'insertion de ville. Ensuite, j'ai créé un tableau des 10 premières villes avec le plus de passages dans une ville et je l'ai affiché. Ce traitement a été difficile, surtout à cause des erreurs de segmentation auxquelles j'ai pu résoudre avec l'aide de Chatgpt même si des fois lui non plus ne savait pas comment faire. Ces problèmes étaient principalement liés à des allocations de pointeurs, complexes à résoudre, ce qui a pris beaucoup de temps.

Le traitement S m'a semblé plus facile, car il était très similaire au traitement T. J'ai simplement dû calculer la moyenne de la somme dans la fonction d'insertion de route, sans avoir besoin de filtrer les villes traversées deux fois par trajet.

Cependant, j'ai rencontré des difficultés lors de l'assemblage des deux traitements, car j'avais initialement conçu les structures de manière similaire, ce qui a entraîné des erreurs de segmentation. J'ai perdu du temps à résoudre ce problème en mettant 2 champs dans la structure Arbre, un pour la ville et un pour le trajet. En fonction du traitement T ou S un seul de ces champs est utilisé. Cela m'a permis de ne pas dupliquer les fonctions liées à l'arbre.

En ce qui concerne les problèmes dans les traitements, le fichier doit obligatoirement s'appeler data.csv pour que le programme fonctionne. Le traitement T est assez long, prenant environ 180 secondes, probablement en raison de nombreuses allocations. Au début, j'avais fait des tableaux avec des valeurs énormes, et le traitement était plus rapide, mais cela gaspillait de la mémoire. À la fin du traitement T, pour que les 10 villes soient classées par ordre croissant, nous avons dû utiliser la commande "sort" car mes tentatives avec qsort dans le programme C ont conduit à des erreurs de segmentation que je n'ai pas réussi à résoudre.

Dans les derniers jours on s'est rejoint 2 fois en salle infos pour travailler le projet. La première fois c'était pour assembler tout ce que l'on avait fait chez nous et la deuxième fois pour vérifier que tout marchait bien.

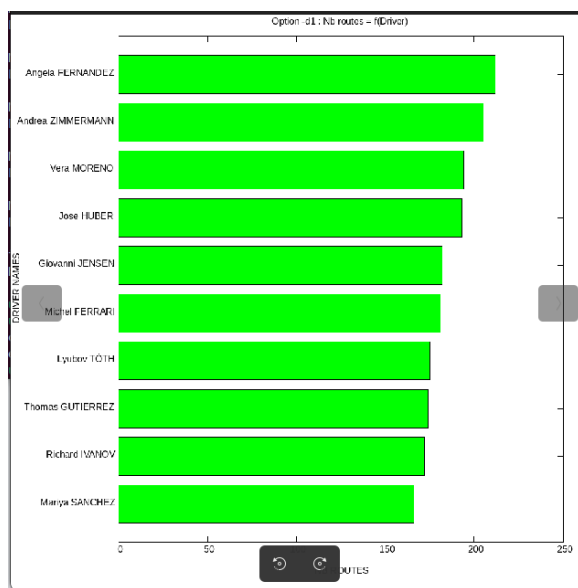
ANNEXES :

Traitement T : moment ou je n'avais pas encore bien tout filtré puisque on peut voir que l'on passe 113 fois a STE COLOMBE et que mon filtrage indique qui passe 111fois beaucoup d'erreur de segmentation a cause de cela.

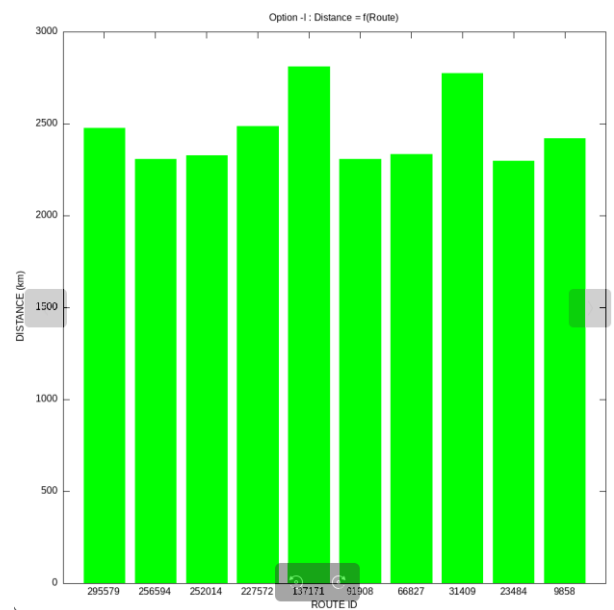
```
MAIN      STE COLOMBE 139462 113
INSERTAVL STE COLOMBE 139462 111

Affichage des resultats
[MARSEILLE,3400,587]-> [ST SAUVEUR,3043,350]-> [STE COLOMBE,2789,111]-> [SALLES,2151,380]-> [LE PIN,2103,27]
-> [BEAULIEU,1998,25]-> [MONS,1942,74]-> [ST MICHEL,1909,167]-> [ANDILLY,1900,407]-> [ST LEGER,1820,44]->
beaulieu-beaulieu: 116 actual Box: 5
```

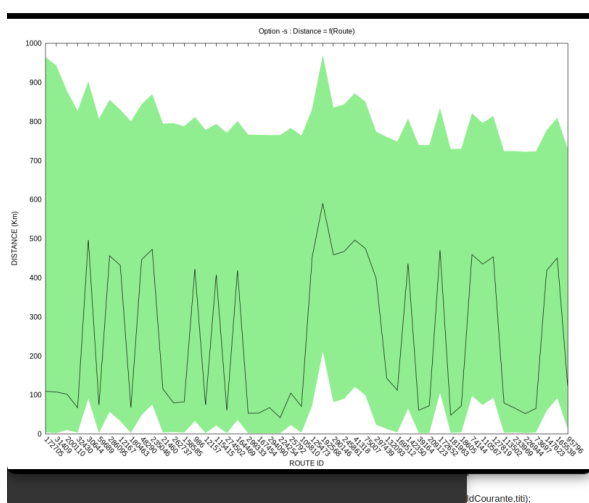
Résultat traitement d1 :



Résultat traitement l :



Résultat traitement s :



Résultat traitement t :

