

Compte rendu du Projet LU2IN006 - 2020-2021

Numéro du groupe de TD/TME : Groupe 7

Nom : MILIVOJEVIC	Nom : AOUCHICHE
Prénom : Nicolas	Prénom : Juba
N° étudiant : 28606874	N° étudiant : 28610319

Réponse aux questions:

Exercice 01:

Q1.1:

Création du fichier Chaines.c avec la fonction Chaines* lectureChaine(FILE* f);, inclus dans le fichier (***voir code, ligne 22-86***)

Q1.2 : Implémentation de la fonction void ecrireChaine(Chaines *C, FILE *f);, au fichier Chaines.c.
Création du fichier ChainesMain.c (***voir code, ligne 10-23***)

Q1.3 : Implémentation de la fonction void afficheChaineSVG(Chaines *C, char* nomInstance);, qui permet de afficher la Chaîne avec la SVG.
Affichage de la chaîne sous forme d'un dessin dans le fichier "affichageSVG.txt ;

Q1.4 : Implémentation des fonctions suivants dans le fichier Chaine.c :
--- double longueurChaine(CellChaine *c); (***voir code, ligne 149-167***)
--- double longueurTotale(Chaines *C); (***voir code, ligne 168-176***)

Q1.5 : Implémentation de la fonction int comptePointsTotal(Chaines* C) dans le fichier Chaine.c
(***voir code, ligne 177-188***)

Exercice 02:

Q2.1 : Création du fichier Reseau.c et implémentation de la fonction
Noeud* rechercheCreeNoeudListe(Reseau *R, double x, double y); dans celui-ci (***voir code, ligne 44-68***)

Q2.2 : Implémentation de la fonction Reseau* reconstitueReseauListe(Chaines *C); dans le fichier Reseau.c
(***voir code, ligne 94-141***)

Q2.3 : Création du programme main dans le fichier ReseauMain.c (***voir code, ligne 12-56***)

Exercice 03:

Q3.1 : Implémentation des fonction suivantes dans le fichier Reseau.c :
— nbCommodites(Reseau *R); (***voir code, ligne 216-223***)
— int nbLiaisons(Reseau *R); (***voir code, ligne 200-214***)

Q3.2 : Implémentation de la fonction void ecrireReseau(Reseau *R, FILE *f); dans le fichier Reseau.c pour le fichier 00014_burma.res. (***voir code, ligne 165-199***)

Q3.3 : Récupération de la fonction void afficheReseauSVG(Reseau*R, char* nomInstance); dans le fichier affichageReseau.txt et testé dans le fichier ChaineMain.c. (***voir code, ligne 26-33***)

Exercice 04:

Q4.1 : Création d'une structure TableHachage dans le fichier Hachage.h qui est la suivante : (***voir code, ligne 8-13***)

```
typedef struct table{  
    int m;  
    int n;  
    CellNoeud** noeuds;  
} TableHachage;
```

Q4.2 : Création des clés par la fonction double clef(double x,double y) ; dans le fichier Hachage.c (***voir code, ligne 8-12***)
Grâce à la fonction clé, nous pouvons prendre une clé échantillonnage pour la fonction de Hachage, donc elle est approprié mais elle est seulement satisfaisante seulement avec la fonction de Hachage.

Q4.3 : Implémentation de la fonction int fonctionHachage(int k, int M) ; dans le fichier Hachage.c pour hachée les données entrant. (***voir code, ligne 13-19***)

Q4.4 : Implémentation de la fonction Noeud* rechercheCreeNoeudHachage(Reseau* R, TableHachage*H, double x, double y); dans la fonction Hachage.c pour recherché,créer un nœud de Hachage. (***voir code, ligne 40-75***)

Q4.5 : Implémentation de la fonction Reseau* reconstitueReseauHachage(Chaines *C, int M); dans le fichier Hachage.c pour reconstituer un Réseau de hachage (***voir code, ligne 76-134***)

Exercice 05:

Q5.1 : Implémentation de la fonction suivante dans le fichier ArbreQuat.c est
void chaineCoordMinMax(Chaines* C, double* xmin, double* ymin, double* xmax, double* ymax); qui permet de calculer les coordonnées minimales et maximales des points dans les chaînes de réseau. (***voir code, ligne 7-22***)

Q5.2 : Implémentation de la fonction suivante dans le fichier ArbreQuat.c est
ArbreQuat* creerArbreQuat(double xc, double yc, double coteX,double coteY); qui permet de créer un ArbreQuat pour les chaînes de Réseau. (***voir code, ligne 23-42***)

Q5.3 : Implémentation de la fonction suivante dans le fichier ArbreQuat.c est
void insererNoeudArbre(Noeud* n, ArbreQuat** a, ArbreQuat*parent); qui permet d'insérer un nœud en trois angles :
— Arbre Vide
— Feuille
— Cellule Interne
(***voir code, ligne 43-96***)

Q5.4 : Implémentation de la fonction suivante dans le fichier ArbreQuat.c est

Noeud* rechercheCreeNoeudArbre(Reseau* R, ArbreQuat** a, ArbreQuat*parent, double x, double y); qui permet de rechercher et créer un nœud Arbre pour les chaînes de Réseau. (***voir code, ligne 97-132***)

Q5.5 : Implémentation de la fonction suivante dans le fichier ArbreQuat.c est

Reseau* reconstruitReseauArbre(Chaines* C); qui permet de reconstituer le Réseau sous forme d'arbre. (***voir code, ligne 133-160***)

Exercice 06:

Q6.1 : Création du fichier ReseauMain.c qui permet d'exécuter les trois reconstitutions avec leurs temps stockée dans le fichier Time_Exec.txt (***voir code, ligne 12-79***)

Q6.2 : Implémentation de la fonction suivante dans le fichier Chaine.c est

Chaines* generationAleatoire(int nbChaines,int nbPointsChaine,int xmax,int ymax) qui permet de créer une Chaine avec des valeurs aléatoires et une taille nbChaines. (***voir code, ligne 189-218***)

Q6.3 : Construction des graphiques avec une chaîne a valeurs aléatoires dans le fichier ReseauMain.c (***voir code, ligne 12-79***)

Q6.4 : On remarque que la vitesse d'exécution est plus rapide pour le Réseau de Hachage que les autres, néanmoins cela est valable seulement sur le long terme vu que l'exécution d'un Réseau de Liste est plus rapide que les autres.

Exercice 07:

Q7.1 : Implémentation de la fonction Graphe* creerGraphe(Reseau* r); dans le fichier Graphe_H.c, qui permet de créer un graphe à partir d'un Réseau. (***voir code, ligne 57-93***)

Q7.2 : Création de la fonction int plusPetit_nbArretes(Reseau* r, Sommet* u, Sommet* v) ; dans le fichier Graphe_H.c, qui permet de calculer le plus petit nombre d'arêtes entre deux sommets u et v. (***voir code, ligne 94-139***)

Q7.3 : Implémentation de la fonction int plusPetit_nbArretes_Liste(Reseau* r, Sommet* u, Sommet* v) ; dans le fichier Graphe_H.c, qui permet de calculer le plus petit nombre d'arêtes entre deux sommets u et v mais avec une liste. (***voir code, ligne 140-187***)

Q7.4 : Implémentation de la fonction int reorganiseReseau(Reseau* r) ; dans le fichier Graphe_H.c, qui permet de réorganisé le Réseau avec un nombre de chaînes limités a gamma et qui soit le plus petit. (***voir code, ligne 188-207***)

Q7.5 : Test de la fonction reorganiseReseau dans le fichier GrapheMain.c (***voir code, ligne 10-26***)

On pourrait proposer comme amélioration pour cette fonction de utiliser les Graphes orientés que les Graphes non-orientés et enlever les doublons.

Description des fichiers:

Les structures utilisées (créé):

Structure TableHachage :

Cette structure permet de créer une variable TableHachage qui stocke la taille, son numéro et un tableau de pointeurs.

Les descriptions rapides des fichiers :

Fichiers Reseau.c et Reseau.h :

Le fichier .c est un fichier contenant le code pour répondre aux exigences de l'Exo2,3 et 6 du Projet et sa fonction main() est dans le fichier ReseauMain.c

Et le fichier Reseau.o est le fichier objet de Reseau.c

Fichier Reseau.h :

Le fichier Reseau.h est un fichier qui donne les fonctions à compléter pour l'Exo2,3 et 6 et il est utilisé par toutes les fichiers.

Fichiers ArbreQuat.c et ArbreQuat.o :

Le fichier .c est un fichier contenant le code pour répondre aux exigences de l'Exo5 du Projet et sa fonction main() est dans le fichier ReseauMain.c

Et le fichier ArbreQuat.o est le fichier objet de ArbreQuat.c

Fichiers Chaine.c et Chaine.o :

Le fichier .c est un fichier contenant le code pour répondre aux exigences de l'Exo1 du Projet, il contient des fonctions qui seront utilisées par la fonction main() qui est dans ChaineMain.c

Et le fichier Chaine.o est le fichier objet de Chaine.c

Fichiers Graphe_H.c et Graphe_H.o :

Le fichier .c est un fichier contenant le code pour répondre aux exigences de l'Exo7 du Projet, et sa fonction main() est dans le fichier GrapheMain.c

Et le fichier Graphe_H.o est le fichier objet de Graphe_H.c

Fichiers Hachage.c et Hachage.o :

Le fichier .c est un fichier contenant le code pour répondre aux exigences de l'Exo4 du Projet, il contient des fonctions qui seront utilisées par la fonction main() qui est dans ReseauMain.c

Et le fichier Hachage.o est le fichier objet de Hachage.c

Fichier ArbreQuat.h :

Le fichier ArbreQuat.h est un fichier qui donne les fonctions à compléter pour l'Exo5 et il est utilisé par le fichier ArbreQuat.c et ReseauMain.c

Fichier Chaine.h :

Le fichier Chaine.h est un fichier qui donne les fonctions à compléter pour l'Exo1 et il est utilisé par toutes les fichiers.

Fichier Graphe_H.h :

Le fichier Graphe_H.h est un fichier qui donne les fonctions a compléter pour l'Exo7 et il est utilisé par le fichier Graphe_H.c et GrapheMain.c

Fichier Hachage.h :

Le fichier Hachage.h est un fichier qui donne les fonctions a compléter pour l'Exo4 et il est utilisé par le fichier Hachage.c et ReseauMain.c

Fichiers ChaineMain.c, ChaineMain.o et ChaineMain :

Le fichier ChaineMain.c est un fichier qui donne un jeu de tests pour testé le programme pour donnée le graphique du Reseau, il inclut tout les fichiers .h

ChaineMain et son exécutable et ChaineMain.o est son fichier objet.

Fichiers GrapheMain.c, GrapheMain.o et GrapheMain :

Le fichier GrapheMain.c est un fichier qui donne un jeu de tests pour testé le programme pour L'Exo6 et 7, il inclut toutes les fichiers .h.

Et le fichier GrapheMain est son exécutable et GrapheMain.o est un fichier objet de GrapheMain.c

Fichiers ReseauMain.c, ReseauMain.o et ReseauMain :

Le fichier ReseauMain.c est un fichier qui donne un jeu de tests pour testé le programme pour L'Exo4 et 5, il inclut toutes les fichiers .h.

Et le fichier ReseauMain est son exécutable et ReseauMain.o est un fichier objet de ReseauMain.c

Fichier Makefile:

Le fichier Makefile est un fichier qui donne des instructions au terminal lors de l'exécution de la commande **make**, il compile toutes les fichiers pour le Projet

Fichiers Time_Exec.txt :

C'est le fichier de réceptions aux données du programmes, donc elle ne sont jamais ouverte Il est assignée aux fichier ReseauMain.

Fichiers .res et .cha :

Ces fichiers sont des fichiers de données comportant des données pour le Projet. Ce sont des exemples.

Fichiers commande.txt :

Ce fichiers est un fichier de commandes pour les données du programmes, donc elle est indispensable

Ils sont assignées aux fichiers suivantes :

- commande.txt pour le fichier Time_Exec.txt pour les fichiers images

Fichier ReseauMain_pic.png:

Ce fichiers est une images montrons un graphe reliant aux fichiers .txt (commande, et Time_Exec)

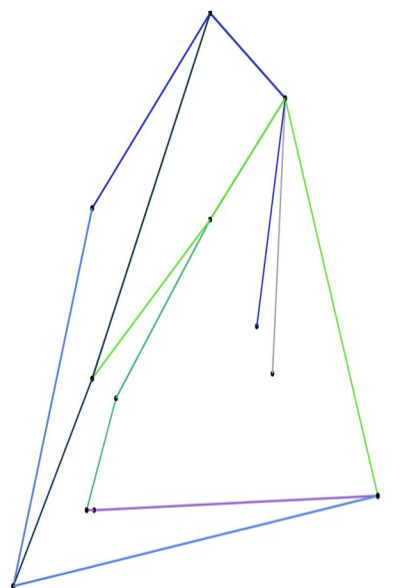
Il est incluse dans aucun programme ou autres fichiers mais il est visible a la page suivante

Autres fichiers :

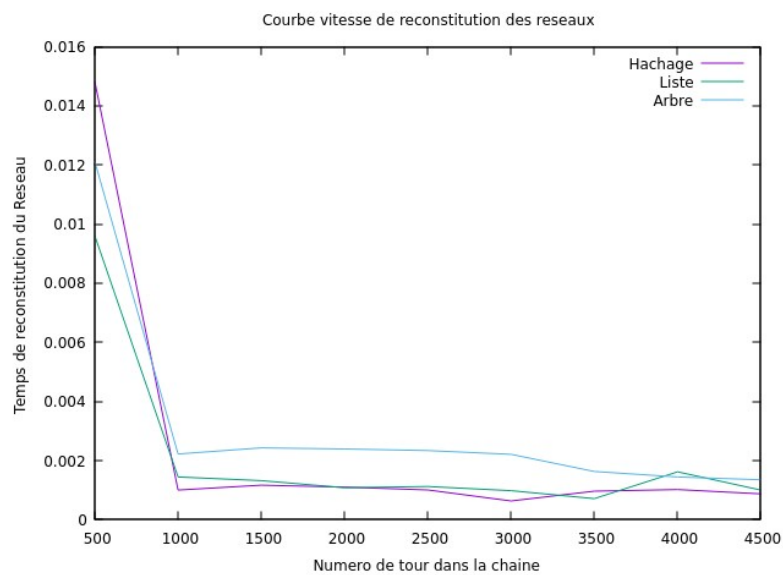
Les autres fichiers sont des fichiers données ou volontairement donnée mais qui sont sans importance

Sauf pour l'exo 1 et 3 pour SVGwriter(.o,.c et .h) et l'exo 7 pour Struct_File et Struct_Liste (.o,.c et .h)

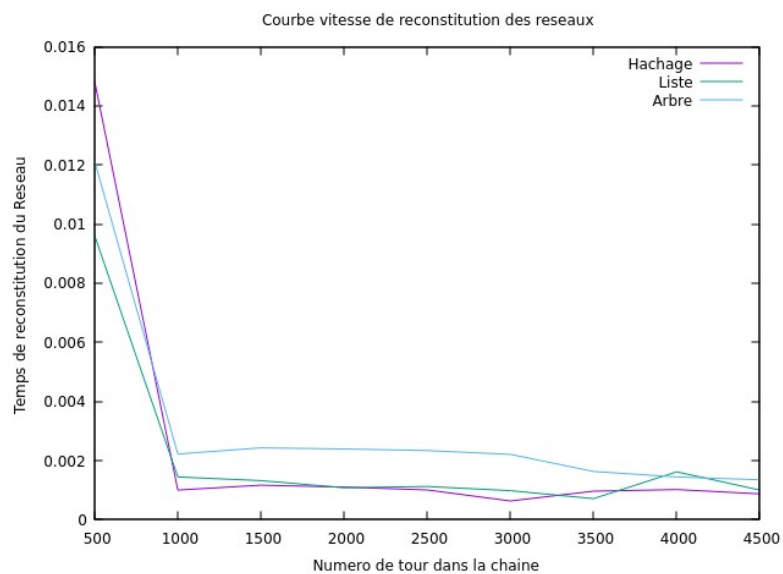
Graphes des programmes :



EX1_3



EX6_1



EX6_3