1. En tant qu’utilisateur je veux, en entrant un fichier source, afficher un graphe.

Format du fichier : .txt

Reader : lecture du fichier.

Parser : vérification de l’intégrité du fichier (tests unitaires).

Erreurs : nœud/relation non unique, fichier mal écrit. Affichage quand même avec erreurs.

Création de graphe : structure à utiliser (voir API graphe -> réponses aux attentes) + tests unitaires.

Affichage de graphe : Listing de tous les liens d’un nœud dans un premier temps.

Complexité lecture du fichier : 2/10

Durée lecture du fichier : 2 heures

Complexité parser + erreurs : 4/10

Durée parser + erreurs : 8 heures

Complexité création du graphe + affichage : 6/10

Durée création du graphe + affichage : 3 heures

2. En tant qu’utilisateur je veux, en entrant un nœud de départ et un lien obtenir tous les nœuds de sortie.

Noeud de départ + lien en paramètres.

Affichage du noeud de sortie.

Complexité de la méthode : 2/10

Durée méthode : 1h

3. En tant qu’utilisateur je veux pouvoir choisir un mode de parcours.

Noeud de départ en paramètre.

Affichage d’un graphe avec le mode de parcours sélectionné.

Complexité de la méthode : 3/10

Durée méthode : 2h

4. En tant qu’utilisateur je veux pouvoir choisir un niveau de parcours.

Noeud de départ en paramètre.

Affichage d’un graphe avec le mode du niveau de parcours sélectionné.

Complexité de la méthode : 2/10

Durée méthode : 1h

5. En tant qu’utilisateur je veux pouvoir choisir l’unicité.

Noeud de départ en paramètre.

Affichage d’un graphe avec l’unicité sélectionnée.

Complexité de la méthode : 2/10

Durée méthode : 1h

6. En tant qu’utilisateur je veux pouvoir choisir les liens à parcourir.

Noeud de départ en paramètre.

Complexité de la méthode : 2/10

Durée méthode : 1h

// En tant qu’utilisateur je veux que l’application survive à une montée en charge