SAE Graphe

BAUDOIN Mathieu
GRECO Lucas
S2A

2) Représentation d’un graphe

Dans cette partie nous avons créé la classe Arc, l’interface graphique et la classe GrapheListe.

Nous avons fait des test qui porte sur :

- Création d’un Arc
- Ajout d’un Arc à une liste de type Arcs
- Ajout d’un Arc qui n'existait pas à une GraphListe
- Récupérer l’Arc suivant dans une GrapheListe

3) Calcul du plus court chemin par point fixe

Dans cette partie nous avons fait la class pour l'algorithm de Bellman Ford et le main pour l’exécuter

Question 8)
Algo logique :

```
fonction pointFixe(Graphe g, Noeud depart)

    Valeur valeurs

    liste_noeud <- g.getNoeud()

    pour n dans noeud faire

            L(n)<-infini

    fpour

   L(depart)<-0

    boolean iter_changed <- vrai

    tant que iter_changed
```

```
iter_changed <- false

pour n dans liste_noeud faire

        pour arc dans g.suivant(n).getArcs // arc de type Arc

                distance <- arc.getCout
                dest <- arc.getDest

                old_distance <- L(dest)
                new_distance <- L(n)+distance

                si (new_distance < old_distance) alors

                        parent(dest)<-n
                        L(dest)<-new_distance

                        iter_changed <- true
                fsi
        fpour
fpour
```

Les tests que nous avon fait était les suivant :

- Assigner un parent à un sommet dans Valeur
- Retrouver le parent d'un sommet dans une Valeur
- Test sur l'algorithme de Bellman

4) Calcul du meilleur chemin par Dijkstra

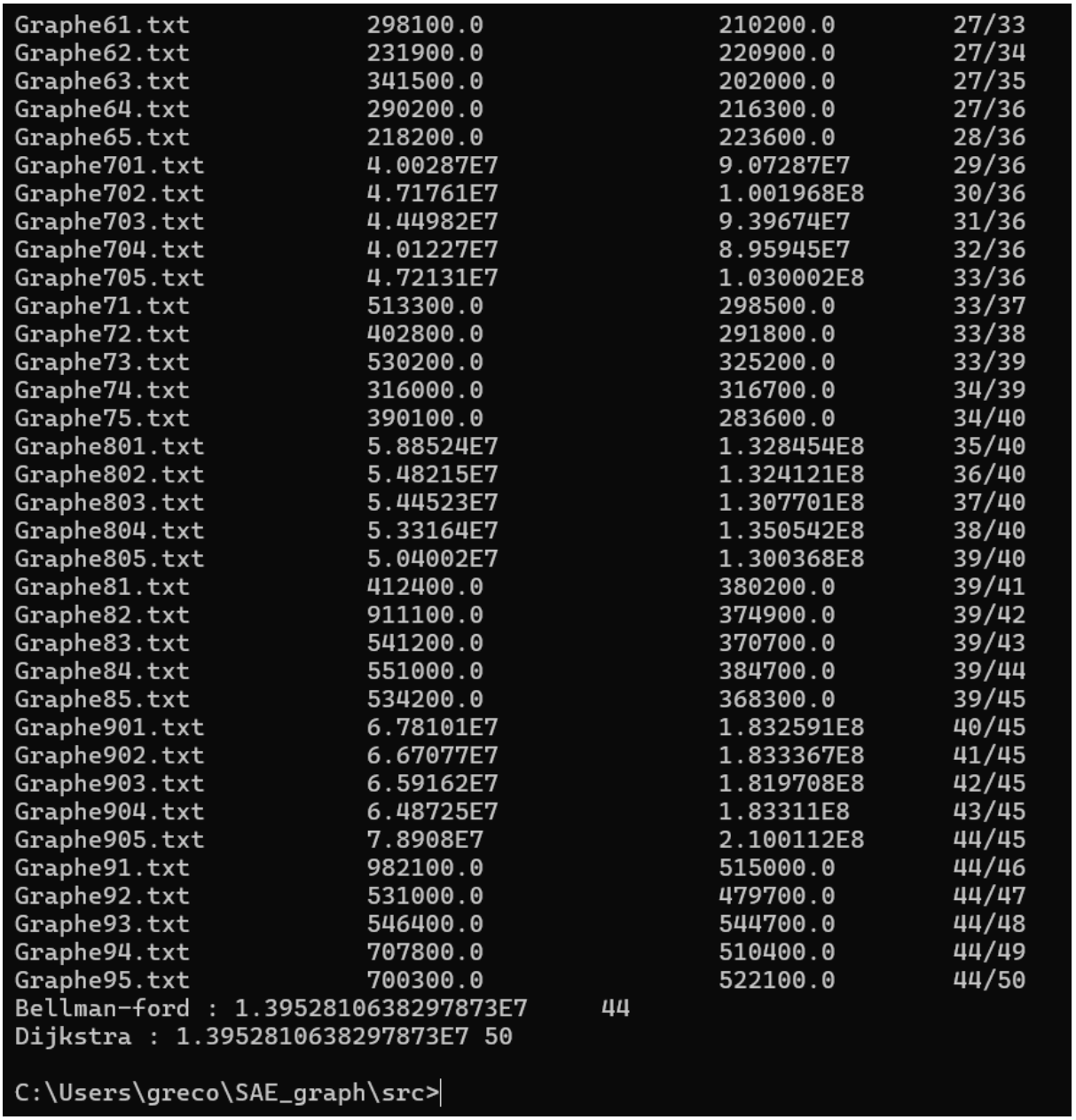
Dans cette partie nous avons fait la classe pour l'algorithme de Dijkstra

Nous avons fait un test sur l'algorithme de Dijkstra

5) Validation et expérimentation

Dans la classe Comparaison.java, nous pouvons comparer les 2 algorithmes.

Voici les résultats sur une de nos machines.



On observe que, sur les graphes donnés, les deux algorithmes se valent au niveau de la moyenne des temps d'exécution. Cependant, l'algorithme de Dijkstra est plus performant dans 50 cas sur 94. Il a aussi un écart-type plus petit.

Conclusion)

Dans cette SAE, nous avons appris à implémenter des algorithmes de recherche de chemin et à les comparer.