

Foulques Géraud

Rapport de projet de Programation Fonctionnelle

19 janvier 2018

I – Introduction

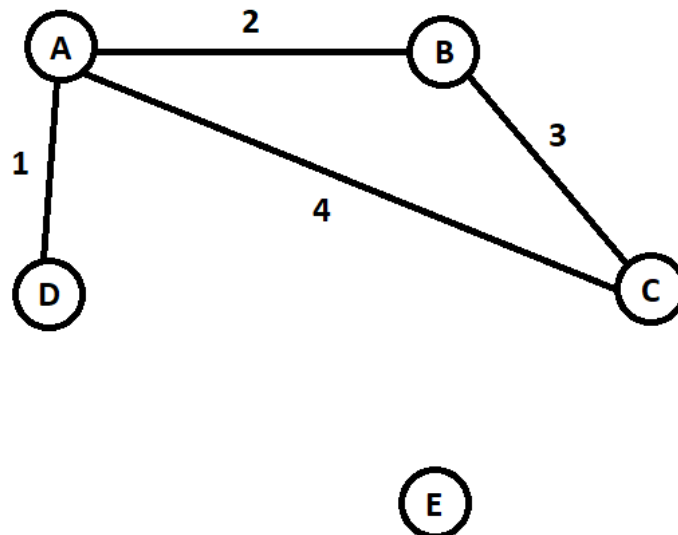
Le présent rapport a pour objet d'expliquer les différentes démarches réalisées dans le cadre du projet de programmation fonctionnelle de l'année 2018. Il comprendra une partie sur le modèle utilisé pour implémenter les graphes non orientés, une partie sur le traitement des données par l'algorithme de Dijkstra et enfin une partie concernant diverses remarques sur le code.

Le travail attendu était en trois parties, c'est notre réalisation de la première que nous présentons ici.

II – Implémentation de graphes

Pour implémenter les graphes, j'ai pris le parti de faire des tableaux associatifs de tableaux associatifs. Tout graphe de sommets G et d'arcs A est représenté par un tableau, qui à chaque valeur s de G associe un sous-tableau, qui, à chaque valeur s' de G telle que $s-s'$ est un arc de A , associe $p(s-s')$ le poids de cet arc.

Par exemple pour le graphe suivant :



Sera implémenté comme suit :

```
Graphe : | A => | B => 2
          |      | C => 4
          |      | D => 1
          |      |
          | B => | A => 2
          |      | C => 3
          |      |
          | C => | B => 3
          |      | A => 4
          |      |
          | D => | A => 1
          |      |
          | E => Sommet.empty
          |
```

Où Sommet.empty est obtenu grâce au foncteur Map.Make, le module String étant passé en argument. Ce foncteur nous renvoie un module de tableaux associatifs dont les clefs sont des chaînes de caractères et dont le type correspondant à un tableau vide est Sommet.empty .