

MTH6412b: Projet voyageur de commerce

Phase 1

Auteur: El Hadji Abdou Aziz NDIAYE (1879468)

Le code source se trouve à l'adresse: [repertoire GitHub](#)

Importation du code

```
Base.show
```

```
Main.workspace28.Node
```

```
Main.workspace28.Edge
```

```
Main.workspace2.add_edge!
```

```
Main.workspace118.plot_graph
```

```
Main.workspace118.main
```

Type Edge

Le type `edge` permet de représenter les arêtes d'un graph. Chaque objet de type *Edge* est composé de trois champs: `name`, `nodes` et `weight`.

Le champ `name` est une chaîne de caractère de type `String` comportant le nom de l'arête.

Le champ `nodes` est un vecteur de deux éléments de type *Node* correspondant aux noeuds de l'arête.

Le champ `weight` est un nombre qui représente le poids de l'arête.

Exemple:

- Création d'une arête:

```
• begin
•   node1 = Node("Joe", 3.14)
•   node2 = Node("Steve", exp(1))
•   weight_ = 250.0
•   edge = Edge("Joe2Steve",[node1,node2],weight_)
•   nothing
```

- Affichage de l'arête:

Edge Joe2Steve, weight: 250.0, composed by nodes: Joe and Steve

```
• with_terminal() do
•   show(edge)
```

Extension du type Graph

Les arêtes sont stockées dans un nouveau champ `edges` qui est un vecteur dont les éléments sont de type *Edge*.

Exemple:

- Création du graphe:

```

• begin
•   nodea = Node("Joe", 3.14)
•   nodeb = Node("Steve", exp(1))
•   nodec = Node("Jill", 4.12)
•   edge1 = Edge("E1", [nodea, nodea], 51.0)
•   edge2 = Edge("E2", [nodeb, nodeb], 22.0)
•   edge3 = Edge("E3", [nodec, nodea], 17.0)
•   G = Graph("Ick", [nodea, nodeb, nodec], [edge1, edge2, edge3])
•   nothing

```

- Affichage du graphe:

```

Graph Ick has 3 nodes and 3 edges.
Node Joe, data: 3.14
Node Steve, data: 2.718281828459045
Node Jill, data: 4.12
Edge E1, weight: 51.0, composed by nodes: Joe and Joe
Edge E2, weight: 22.0, composed by nodes: Steve and Steve
Edge E3, weight: 17.0, composed by nodes: Jill and Joe

```

```

• with_terminal() do
•   show(G)

```

Lecture des poids des arêtes:

La méthode `read_edges` a été adaptée afin de lire les poids des objet

Programme principal

La fonction `stsp_to_graph` permet de lire un fichier `stsp` et de stocker les données dans une structure de type *Graph*.

La fonction `main` permet de lire l'ensemble des fichier contenus dans le repertoire *intances/stsp*.

- Exemple de lancement de la fonction `stsp_to_graph`:

File: bayg29.tsp
Reading of header : OK
Reading of nodes : OK
Reading of edges : OK

```
• with_terminal() do
•   filename = "../instances/stsp/bayg29.tsp"
•   showgraph = false;
•   plotgraph = false;
•   graph_ =
  stsp_to_graph(filename;show_graph_flag=showgraph,plot_graph_flag=plotgraph);
•   nothing
```

- Exemple de lancement de la fonction main :

```
File: bayg29.tsp
Reading of header : OK
Reading of nodes : OK
Reading of edges : OK
File: bays29.tsp
Reading of header : OK
Reading of nodes : OK
Reading of edges : OK
File: brazil58.tsp
Reading of header : OK
Reading of nodes : OK
Reading of edges : OK
File: brg180.tsp
Reading of header : OK
Reading of nodes : OK
Reading of edges : OK
File: dantzig42.tsp
Reading of header : OK
Reading of nodes : OK
```

- `with_terminal()` `do`
- `main()`