

# Ecole polytechnique de Montréal

---

## MTH6412B Session Automne 2021

### Rapport de projet

---

**Pierre Mordant & Mahamadou Sarité**

### Phase 1

Dans cette première phase du projet sur le problème du voyageur de commerce symétrique, nous définissons des structures de données de base adéquates. Nous procédons à des modifications des codes de démarrages et à proposer:

1. Un type Edge pour présenter les arêtes d'un graphe.
2. Etendre le type Graph afin qu'un graphe contienne ses arêtes.
3. Etendre la méthode d'affichage show d'un objet de type Graph afin que les arêtes du graphe soient également affichées
4. Etendre la fonction read \_ edges() de read \_ stsp.jl afin de lire les poids des arêtes.
5. Fournir un programme principal main() qui lit une instance de TSP symétrique dont les poids sont donnés au format EXPLICIT et construit un objet de type Graph correspondant.

Le type `Edge` est défini avec 3 attributs comme suit: les deux sommets (`string`, `string`) reliés par l'arête et le poids de cette arête (`Int`). Par exemple, l'arête `Edge(1, 5, 86)` reliant le noeud 1 au noeud 5 ayant pour poids 86 dans le fichier EXPLICIT `bayg29.tsp`. voir la capture ci-dessous pour plus de visualisation. Le type `Graph` renvoie un vecteur des données à format flottant (`Float64`). Ces données sont composées d'un nom `Graph`, une liste de noeuds (`Node`) et une liste d'arêtes (`Edge`). Le type `Graph` a été modifié pour permettre l'inclusion des arêtes.

Le programme principal `main.jl` a été élaboré pour lire une instance TSP symétrique dont les poids sont au format EXPLICIT.

L'utilisateur fait appel à la fonction `create_graph("chemin du fichier EXPLICIT")` dans Julia pour générer le graphe correspondant dans le Terminal. Comme indiqué dans la capture ci-dessous.

```
Graph{Vector{Float64}}("/Users/admin/Documents/PhD Courses/MT6412B/Projet/mth64
1760.0]), Node{Vector{Float64}}("2", [630.0, 1660.0]), Node{Vector{Float64}}("3
Node{Vector{Float64}}("6", [1030.0, 2070.0]), Node{Vector{Float64}}("7", [1650.
ctor{Float64}}("10", [710.0, 1310.0]) ... Node{Vector{Float64}}("20", [590.0, 1
r{Float64}}("23", [1840.0, 1240.0]), Node{Vector{Float64}}("24", [1260.0, 1500.
loat64}}("27", [1460.0, 1420.0]), Node{Vector{Float64}}("28", [1260.0, 1910.0])
139), Edge("1", "5", 86), Edge("1", "6", 60), Edge("1", "7", 220), Edge("1", "8
"27", 120), Edge("25", "28", 205), Edge("25", "29", 270), Edge("26", "27", 213)
```