

# TP A\*

## 1 Description du TP

Vous trouverez sur le serveur une archive `data.zip` qui contient des données géographiques<sup>1</sup> dans deux fichiers :

- `positions.txt` donne les coordonnées en km de certaines grandes villes européennes ;
- `connections.txt` donne les distances routières sur les grands axes entre ces villes.

Ces données sont également représentées de manière graphique à la figure 1 au verso.

Le but de ce TP va être d'utiliser l'algorithme A\* pour trouver des chemins optimaux entre ces villes.

## 2 Heuristiques

Supposons que l'on veuille se rendre à la ville  $B$ . Pour tout noeud  $n$ , on va s'intéresser aux heuristiques suivantes

- $h_0(n) = 0$
- $h_1(n)$  = "la distance entre  $n$  et  $B$  sur l'axe des  $x$ "
- $h_2(n)$  = "la distance entre  $n$  et  $B$  sur l'axe des  $y$ "
- $h_3(n)$  = "la distance à vol d'oiseau entre  $n$  et  $B$ "
- $h_4(n)$  = "la distance de Manhattan entre  $n$  et  $B$ "

Parmi ces heuristiques, lesquelles sont admissibles ?

## 3 A\*

Implémenter, en python, une fonction (ou méthode)

- qui prend en paramètre deux villes et une heuristique,
- qui utilise l'algorithme A\*
- et qui retourne le chemin le plus court entre ces deux villes en indiquant combien de villes ont été "visitées" pour trouver ce chemin optimal ;
- fournir, une interface (graphique ou ligne de commande) pour tester votre implémentation.

Implémenter également les 5 heuristiques ci-dessus<sup>2</sup>.

## 4 Expérimentation

Chercher quelques chemins optimaux à l'aide de votre programme et des différentes heuristiques.

- L'utilisation des différentes heuristiques a-t-elle une influence sur l'*efficacité* de la recherche ? (en termes du nombre de noeuds visités)
- Pouvez-vous trouver des exemples où l'utilisation de différentes heuristiques donne des *résultats* différents en termes de chemin trouvé ?

---

1. Ces données – très approximatives !

2. Bien entendu, vous devrez également récupérer les données présentes dans les fichiers textes. Ne cherchez pas trop loin, avec quelque chose comme `[l.split() for l in f]` vous avez déjà fait les 3/4 du travail...

- Dans un cas réel, quelle heuristique utiliseriez-vous ? Pourquoi ?
- Aller plus loin : chercher la définition d’heuristique “consistante” ou “monotone”.
  - Quel est son impact sur les performances de l’algorithme  $A^*$  ?
  - Si vous assurez à votre algorithme une heuristique *monotone*, comment pourriez-vous améliorer votre de code ?
  - Parmi les 5 heuristiques ci-dessus, il y en a des *monotones* ? Si non, proposer une heuristique *monotone* pour notre problème du voyageur (pas besoin de l’implémenter)

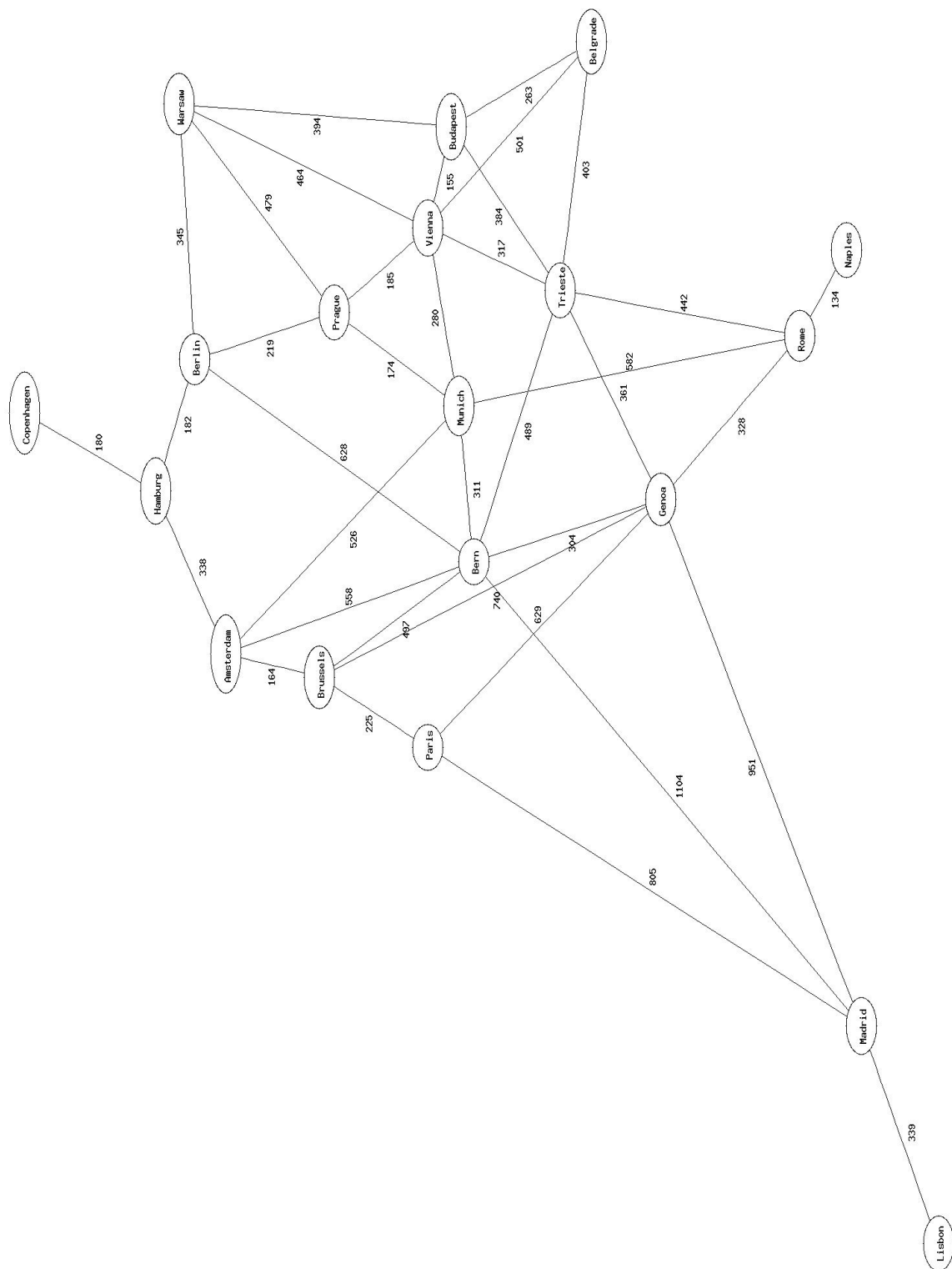


FIGURE 1 – Les villes et distances considérées