

# Algorithmie avancée

Partiel — .decode 2025

## Distance la plus courte (15 points)

On considère un graphe pondéré **sans arêtes négatives** représentant un réseau de routes entre des villes.

Le graphe est donné sous la forme d'un dictionnaire d'adjacence Python :

```
graphe = {  
    "A": [("B", 4), ("C", 2)],  
    "B": [("C", 5), ("D", 10)],  
    "C": [("E", 3)],  
    "D": [("F", 11)],  
    "E": [("D", 4)],  
    "F": []  
}
```

Chaque clé est une ville, et la valeur est une liste de tuples (**voisin**, **poids**) représentant la distance entre deux villes.

**Utilisez Dijkstra pour calculer le chemin le plus court entre 2 villes.**

## Question bonus (5 points)

Transformation en A\*

Vous devez modifier l'algorithme pour qu'il utilise une **heuristique admissible** afin de guider la recherche.

**Implémentez une version simplifiée de l'algorithme A\*, utilisant la distance euclidienne comme heuristique.**