
Devoir 1 : Fer ou Air ? (*théorie des graphes*)

Informations pratiques

Ce devoir est individuel. Il vous est demandé de résoudre ce problème individuellement. Vous êtes autorisé.e, et même encouragé.e, à échanger des idées avec d'autres étudiant.e.s sur la façon d'aborder ce devoir. En revanche, vous devez rédiger votre solution individuellement : ne partagez pas votre production. Nous serons intransigeant.e.s si nous observons des similitudes dans votre code, lequel passera dans les logiciels anti-plagiat de Gradescope. Pour les consignes concernant l'usage de logiciels d'intelligence artificielle, veuillez vous référer au Moodle du cours.

Echéance : le 18 Octobre 2023 à 23h59.

Soumission

Vous remettrez votre production via les étapes suivantes :

1. Sur <https://www.gradescope.com/>, se connecter via le bouton “Log In” en haut à droite.
2. Cliquer sur “School Credentials” en bas à gauche.
3. Cliquer sur “UCLouvain Username” près du bas de la liste des universités.
4. Vous serez redirigés vers la page d'identification UCLouvain, sur laquelle vous vous connectez comme d'habitude.
5. Si vous êtes inscrits sur la page Moodle du cours, vous devriez voir apparaître LEPL1108 dans la liste de vos cours. Si ce n'est pas le cas, vous devriez pouvoir rejoindre le cours en vous servant du code **6GGV63**.
6. Vous avez un fichier “squelette” à compléter `devoir1.py` qui sera fourni sur Moodle. Pour ce devoir, vous n'avez qu'un fichier à soumettre votre code Python nommé `devoir1.py`. Attention, si vous nommez votre fichier autrement, il échouera aux tests de Gradescope.

NB :

- Vous devez respecter le modèle du fichier “squelette” fourni. Vous risquez sinon d'avoir des problèmes de lecture du code par Gradescope. Pour information, Gradescope tourne sur la version 3.6.9 de Python et seules les bibliothèques *numpy* et *collections* sont autorisées pour ce devoir.
- La section “Exemple” ci-dessous contient un test public avec sa réponse. Ce test vous permet d'évaluer la qualité de votre fonction. Il faut savoir que Gradescope corrigera bien le test public mais sa pondération est de 0 point sur 20 de la note finale.
- La note finale sera évaluée sur base de tests secrets. Gradescope vous indiquera uniquement si votre code réussit le test public ; le reste étant caché. La réponse et la note finale des tests secrets restent cachés, ce qui empêche de “hard-coder” les réponses finales. Si votre code fonctionne correctement sur le tests public, il ne devrait pas avoir de soucis avec les tests secrets.

Contexte

Vous êtes coordinateur.trice d'une agence de voyage spécialisée en voyages européens. Vous aimeriez mettre à jour vos offres pour proposer des voyages écoresponsable à vos clients.

Votre objectif : comparer les durées de voyage inter-européen entre l'option ferroviaire ou aérienne.

Pour ce faire, vous vous lancez dans une collecte de données extensive pour cartographier le réseaux ferroviaire européen ainsi qu'aller chercher les 23 villes les plus attractives en Europe centrale. Vos données collectées sur le réseaux ferroviaire européen sont représentées dans la Figure 1 sous la forme d'un graphe connexe et les durées des trajets constituant la pondération des arêtes du graphe sont représentées dans la Table 1. Vous vous lancez également dans la collecte de données sur tous les vols direct partant de et arrivant à chacune des 23 villes européennes. Chacune des 23 villes est considérée comme ayant un vol direct vers les 22 autres.

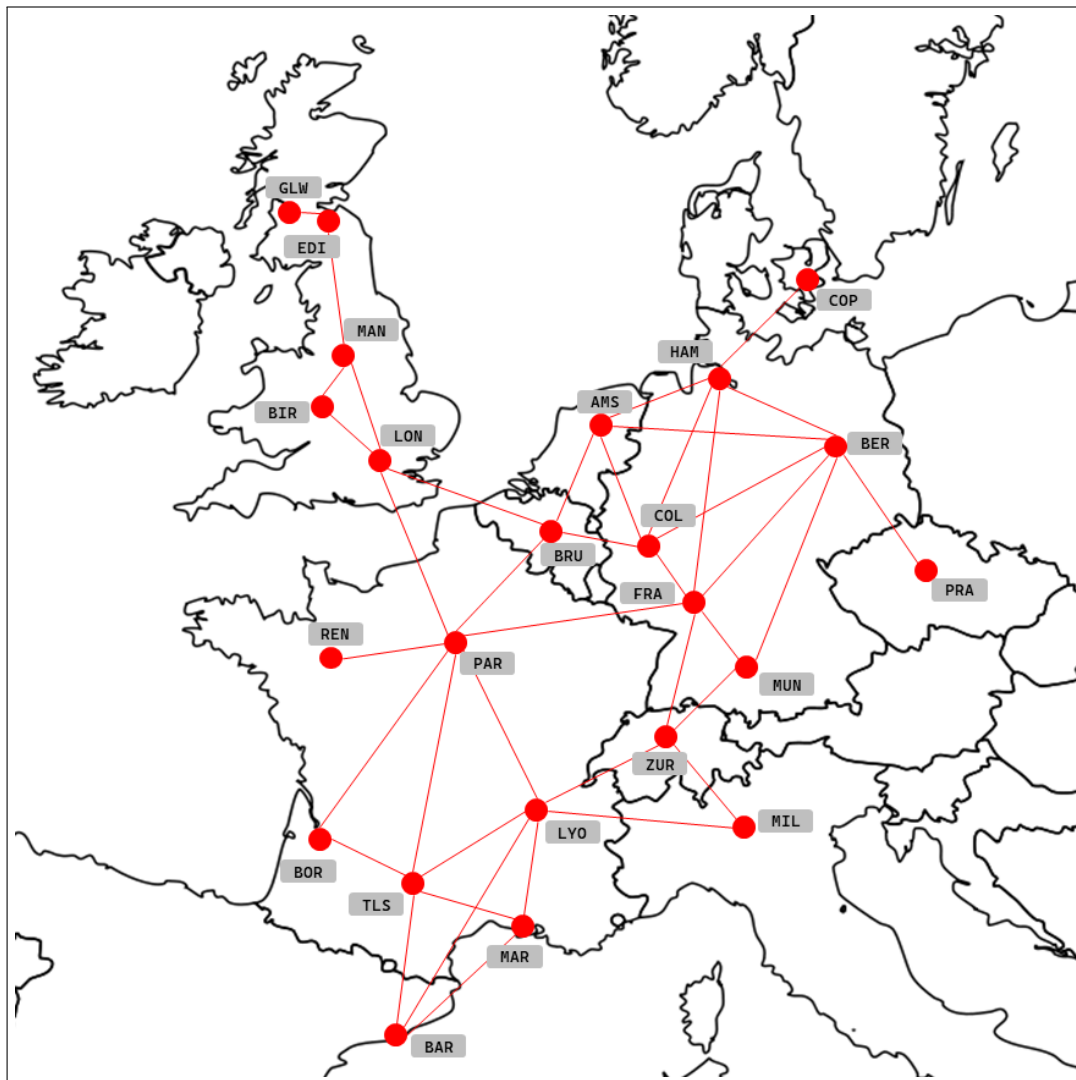


FIGURE 1 – Graphe représentant le réseau ferroviaire européen

Trajets	Durées
BRU-PAR	80
AMS-COL	180
FRA-ZUR	235
TLS-BAR	195
BRU-AMS	95
AMS-HAM	300
BER-MUN	255
TLS-MAR	225
BRU-LON	120
COL-HAM	215
BER-HAM	175
LON-BIR	95
BRU-COL	105
COL-FRA	60
BER-PRA	265
LON-MAN	140
PAR-BOR	120
COL-BER	280
MUN-ZUR	210
MAN-BIR	80
PAR-LYO	110
LYO-MAR	110
MIL-LYO	270
MAN-EDI	190
PAR-FRA	230
LYO-ZUR	240
MIL-ZUR	240
EDI-GLW	50
PAR-LON	140
FRA-HAM	280
LYO-BAR	330
LYO-TLS	250
PAR-REN	90
FRA-BER	275
BAR-MAR	285
HAM-COP	280
AMS-BER	385
FRA-MUN	200
BOR-TLS	120
PAR-TLS	260

TABLE 1 – Liste des trajets ferroviaires européens et leurs durées en minutes des 23 villes européennes

Énoncé

Il vous est demandé d'atteindre votre objectif en rédigeant une fonction en Python, s'inspirant des algorithmes de Bellman-Ford et de Dijkstra, dont la signature est donnée ci-dessous :

```
1 def devoir1(trajets_train, durees_train, ville_depart, durees_voldirect, sacrifice
2 ):
3     # to do
4
5     return ville, min_distances_train, min_distances_avion, count_train
```

Pour une comparaison équitable, nous devons tenir compte de la durée du trajet de centre-ville à centre-ville et pénalise le temps perdu comme indiqué à la Figure 2. Nous devons donc tenir compte des pénalités de temps suivantes :

- Le temps moyen pour rejoindre l'aéroport depuis le centre ville et vice versa est estimé à 30 minutes étant donné que les aéroports sont généralement situés en dehors de la ville. Il faut donc ajouter une pénalité de 60 minutes à chaque vol dans votre fonction.
- Les compagnies aériennes recommandent d'être deux heures à l'avance à l'aéroport avant le départ. Il faut donc ajouter 120 minutes de pénalité supplémentaire à chaque vol dans votre fonction.
- Le temps moyen pour atteindre la gare centrale depuis le centre ville et vice versa est estimé à 15 minutes étant donné que les gares centrales sont généralement proches du centre ville. Vous devez donc ajouter une pénalité de 30 minutes à chaque trajet en train dans votre fonction.
- Si un transfert est nécessaire pendant votre voyage en train, vous devrez ajouter une pénalité de 15 minutes pour chaque transfert dans votre fonction.

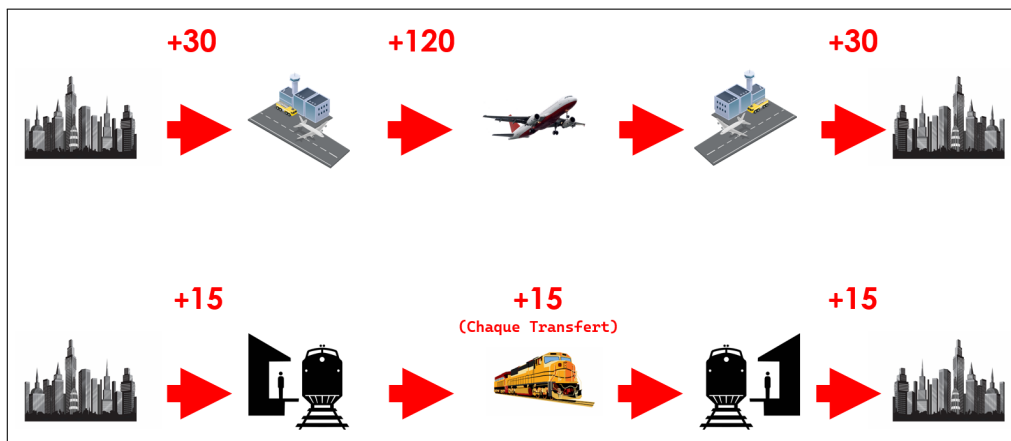


FIGURE 2 – Pénalité de temps perdu du trajet de centre-ville à centre-ville

Inputs

- **trajets_train** est une liste de strings. Chaque string de la liste représente une arête non-dirigée dans le graphe. Ex : ["BRU-PAR", "AMS-COL", "FRA-ZUR", "TLS-BAR", ..., "PAR-TLS"].
- **durees_train** est une liste d'entiers. Chaque entier représente la pondération de l'arête correspondante dans le graphe. Ex : [80, 180, 235, 195, ..., 260], signifiant que le trajet "BRU-PAR" prend 80 minutes, le trajet "AMS-COL" prend 180 minutes, et ainsi de suite.
- **ville_depart** est un string de trois caractères en lettres majuscules qui représente la ville de départ choisie dans notre évaluation. Ex : 'BRU'.
- **durees_voldirect** est une liste d'entiers. Chaque entier représente la durée d'un vol direct en provenance de la ville de départ sélectionnée vers les 22 villes restantes. La liste est triée par ordre alphabétique. Ex : ['AMS', 'BAR', 'BER', 'BIR', ..., 'ZUR'].
- **sacrifice** est un entier qui représente la durée en minutes qu'une personne est prête à sacrifier pour prendre le train au lieu de l'avion. Ex : 60, signifiant que la personne est prête à ajouter 60 minutes sur la durée globale de son trajet pour opter pour une option de transport plus durable.

Output

- **ville** est une liste de strings. Chaque string de la liste représente une des 22 villes de destination (ville de départ exclus) dans le graphe. Ex : ['AMS', 'BAR', 'BER', 'BIR', 'BOR', 'COL', ..., 'TLS', 'ZUR'].
- **min_distances_train** est une liste d'entiers. Chaque entier représente la durée totale, pénalités incluses, du trajet le plus cours en train pour chacune des 22 villes de destination. Ex : [125, 580, 430, 260, 245, 135, ..., 380, 460].
- **min_distances_avion** est une liste d'entiers. Chaque entier représente la durée totale, pénalités incluses, du trajet le plus cours en avion pour chacune des 22 villes de destination. Ex : [245, 300, 265, 260, 280, 240, ..., 280, 255].
- **count_train** est un entier qui indique le nombre total de villes de destination où le train, prenant en compte le sacrifice de temps qu'une personne est prête à faire pour prendre le train, est une meilleure option que l'avion. Ex : 10, indiquant que le train est une meilleure option que l'avion pour 10 des 22 villes de destination, moyennant le fait que la personne est prête à sacrifier 60 minutes supplémentaire à la durée totale de son trajet pour prendre le train.

NB :

- Les noms des villes s'écrivent toujours en trois caractères de lettres majuscules.
- Toutes les durées sont exprimées en minutes.
- Pour **count_train**, si la durée totale, pénalités et sacrifice inclus, en train et en avion est équivalente, la préférence ira au train.

Exemple

Comme exemple, la ville de départ choisie pour le test public sur Gradescope sera la ville de Bruxelles. Nous considérons pour cet exemple qu'une personne est prête à sacrifier 60 minutes pour prendre le train au lieu de l'avion. Les vols directs partant de et arrivant de Bruxelles vers les 22 autres villes européennes est décrit à la Figure 3 et les durées des vols sont listées dans la Table 2.

Pour cet exemple, et en utilisant les tables 1 et 2, les inputs attendus pour la fonction ***devoir1*** sont les suivants :

- `trajets_train` = ['BRU-PAR', 'BRU-AMS', 'BRU-LON', 'BRU-COL', 'PAR-BOR', 'PAR-LYO', 'PAR-FRA', 'PAR-LON', 'PAR-REN', 'AMS-BER', 'AMS-COL', 'AMS-HAM', 'COL-HAM', 'COL-FRA', 'COL-BER', 'LYO-MAR', 'LYO-ZUR', 'FRA-HAM', 'FRA-BER', 'FRA-MUN', 'FRA-ZUR', 'BER-MUN', 'BER-HAM', 'BER-PRA', 'MUN-ZUR', 'MIL-LYO', 'MIL-ZUR', 'LYO-BAR', 'BAR-MAR', 'BOR-TLS', 'TLS-BAR', 'TLS-MAR', 'LON-BIR', 'LON-MAN', 'MAN-BIR', 'MAN-EDI', 'EDI-GLW', 'LYO-TLS', 'HAM-COP', 'PAR-TLS']
- `durees_train` = [80, 95, 120, 105, 120, 110, 230, 140, 90, 385, 180, 300, 215, 60, 280, 110, 240, 280, 275, 200, 235, 255, 175, 265, 210, 270, 240, 330, 285, 120, 195, 225, 95, 140, 80, 190, 50, 250, 280, 260]
- `ville_depart` = 'BRU'
- `durees_voldirect` = [65, 120, 85, 80, 100, 60, 90, 100, 60, 105, 75, 80, 85, 85, 100, 90, 80, 65, 90, 95, 100, 75]
- `sacrifice` = 60

Les outputs attendus pour la fonction ***devoir1*** attendus sont les suivants :

- `ville` → ['AMS', 'BAR', 'BER', 'BIR', 'BOR', 'COL', 'COP', 'EDI', 'FRA', 'GLW', 'HAM', 'LON', 'LYO', 'MAN', 'MAR', 'MIL', 'MUN', 'PAR', 'PRA', 'REN', 'TLS', 'ZUR']
- `min_distances_train` → [125, 580, 430, 260, 245, 135, 660, 510, 210, 575, 365, 150, 235, 305, 360, 520, 425, 110, 710, 215, 380, 460]
- `min_distances_avion` → [245, 300, 265, 260, 280, 240, 270, 280, 240, 285, 255, 260, 265, 265, 280, 270, 260, 245, 270, 275, 280, 255]
- `count_train` → 10

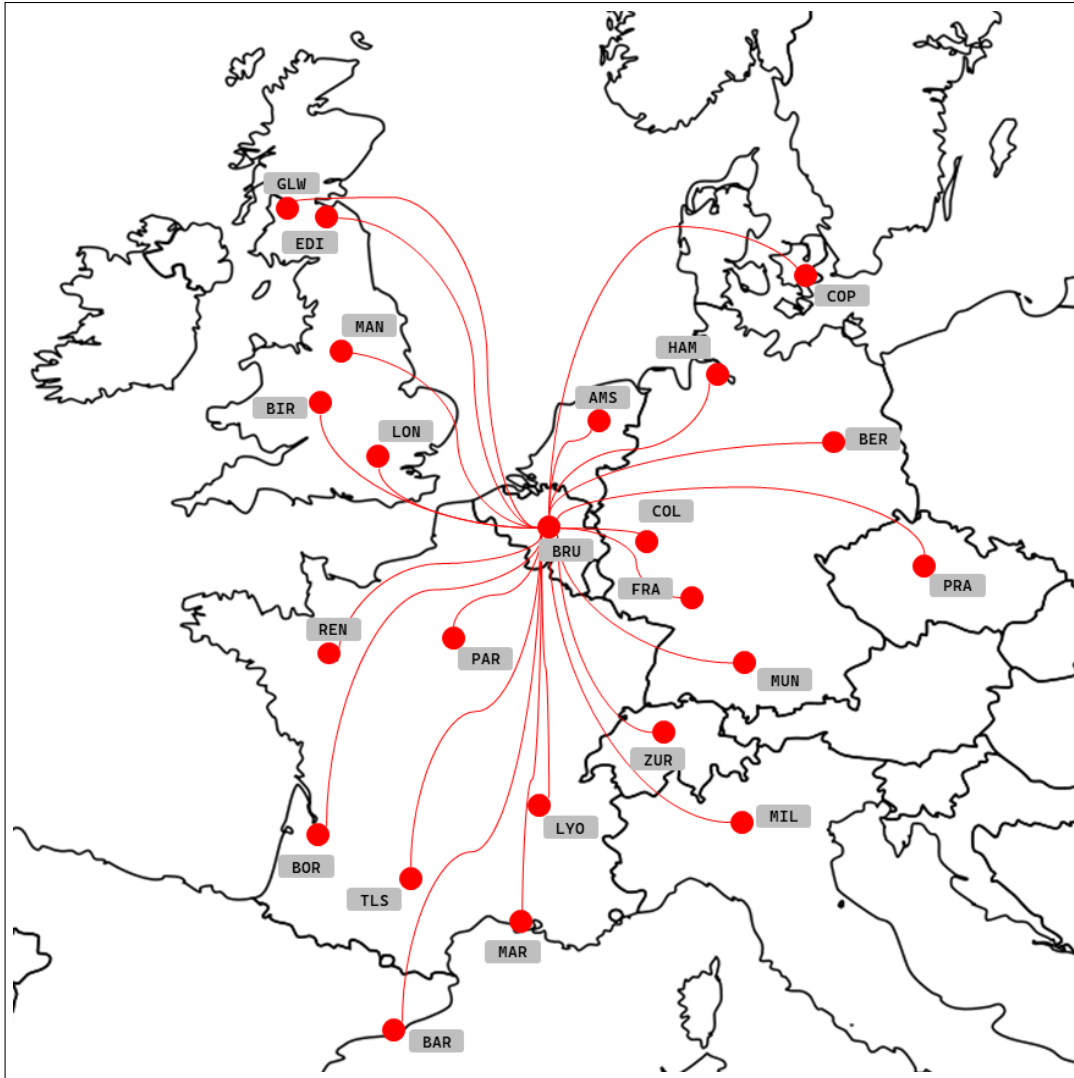


FIGURE 3 – Vols directs partant de et arrivant de Bruxelles vers les 22 autres villes européennes

Destinations	Durées
AMS	65
BAR	120
BER	85
BIR	80
BOR	100
COL	60
COP	90
EDI	100
FRA	60
GLW	405
HAM	75
LON	80
LYO	85
MAN	85
MAR	100
MIL	90
MUN	80
PAR	65
PRA	90
REN	95
TLS	100
ZUR	75

TABLE 2 – Liste des vols directs partant de et arrivant à Bruxelles vers les 22 autres villes européennes et leurs durées en minutes

Le saviez-vous ?

Ce devoir a été inspiré par le travail de fin d'étude d'un ancien étudiant de l'Ecole Polytechnique de Louvain, Damien Lepage (Promotion 2020), en collaboration avec la SNCB.

Son travail, disponible publiquement, montre que pour 6% des trajets entre les principales villes de l'Union Européenne, le temps de trajet porte-à-porte est plus court en train qu'en avion. Cette statistique n'a pas empêché qu'en juin 2017 plus de 4 millions de passagers ont pris un vol pour un trajet où le train était plus rapide. Son étude montre aussi que le train est généralement compétitif par rapport à l'avion jusqu'à une distance de 444 km, et jusqu'à 704 km lorsque le trajet peut être effectué en train à grande vitesse.

Son travail a attiré beaucoup d'attention médiatique avec des articles publiés par exemple dans La Libre, Le Soir et Daily Science.