



M1 MIAGE RO - Consignes projet

Concepteurs : Rumen Andonov, Victor Epain, Arthur Gontier

Organisation

- Vous pouvez réaliser ce projet **seul ou en binôme**.
 - Vous pouvez cependant vous aider entre groupes, mais nous attendons **une rédaction et des programmes particuliers pour chaque groupe**. Des copies de codes ou de rapports seront pénalisés.
- Un rendez vous avec Victor Epain en milieu d'échéance sera prévu pour chaque groupe
- Coefficient du projet pour le calcul de la note totale en RO : 50%
- N'hésitez pas à contacter votre référent de TP si vous avez une question ou un problème
- **Date de rendu : dimanche 31 / 01 / 2021, 23h59**

Mail

- ☐ **Destinataire** : votre référent de TP
- ☐ **Objet** : [M1 MIAGE] RO Projet NOM1 NOM2
- ☐ **Pièce jointe** : archive en zip ou en tar.gz contenant un dossier nommé projet_RO_NOM1_PRENOM1_NOM2_PRENOM2 contenant l'ensemble du programme et le rapport **en pdf**

Programme

- ☐ Implémenté en Python \geq 3.8
- ☐ Un fichier nommé truck_pulp.py et d'autres fichiers python en .py :
- ☐ Contenu minimal requis dans le fichier truck_pulp.py :

```
# -*- coding=utf-8 -*-

"""Module `truck_pulp.py`.

Attention à l'importation de module, explication :
    quand vous lancerez votre test, vous vous trouverez
    un étage précédent dans l'arborescence des fichiers
    par rapport au dossier contenant vos programmes.
    Par conséquent, dans chacun des programmes dans le dossier
    vous devez indiquer un chemin relatif par rapport à votre
    fichier `test.py`
"""

# Pour charger un module dans le dossier du projet
import projet_RO_NOM1_PRENOM1_NOM2_PRENOM2.module as mod
# ...

def solve_truck_problem(file_path):
    # Faire quelque chose ici avec l'argument `file_path`
    # qui est un chemin de fichier

    # ...

    # La fonction retournera :
    # - la valeur de la fonction objectif égale aux bénéfices
    #   de l'entreprise si le problème est résoluble,
    #   sinon `None`. Le type de retour sera un "float" ;
    # - un dictionnaire,
    #   où les clefs sont les routes et les valeurs associées
    #   sont les quantités de marchandises qui les traversent ;
    # - ce que vous voulez en plus si besoin.
    return optval, roads_qty #, ...
```

☐ Un script pour tester votre programme `test.py` .

Ce programme ne se trouve pas dans le dossier du projet à rendre, mais dans le même dossier parent. Vous pouvez tester votre programme en lançant ce fichier :

```
# -*- coding=utf-8 -*-

"""Le programme de test `test.py`."""

from pathlib import Path

# Importer le programme principal qui se trouve dans le dossier du projet
import projet_RO_NOM1_PRENOM1_NOM2_PRENOM2.solve_truck_problem as solve_truck_problem

# La magie de pathlib...
THIS_FILE_DIR = Path(__file__).parent.resolve()
DATA_DIR = THIS_FILE_DIR / 'data'

if __name__ == '__main__':
    # ...
    # instancier `file_path`
    instance_path = DATA_DIR / 'quelquechose.data'
    optval, roads_qty, ... = solve_truck_problem(instance_path)
    # ...
```

- ☐ Tout fichier généré devra se trouver dans un dossier `output_files` qui se trouvera à l'intérieur du même dossier contenant les programmes, **qu'importe l'endroit où on lance le programme**

Conseils

- Utiliser la librairie `pathlib` pour gérer des chemins de fichiers
- Assurez vous de travailler avec des chemins absolus dans vos programmes (vous pouvez partir d'un chemin relatif, et le transformer ensuite grâce au module `pathlib` ...)
- Découpez vos programmes en fonctions, voire même en module
 - attention aux importations de vos modules !
- A moins que vous ne codiez en orienté objet (pas demandé), utilisez des variables globales pour des valeurs fixes partagées (ex : `ALPHA = 'alpha'`). Vous pouvez importer d'un module ces variables globales (ex : `from module import ALPHA`)

La notation prendra en compte la **clarté ainsi que la propreté du code**.

Rapport

- ☐ Le format du rapport sera en **pdf**
- ☐ Le nom du rapport respectera le nom du dossier le contenant, à savoir
`projet_R0_NOM1_PRENOM1_NOM2_PRENOM2.pdf`

Un minimum de question sera posé. Nous nous attendons cependant à des initiatives de votre part concernant la présentation de votre travail. La notation prendra notamment en compte :

- La structure du rapport
- La présentation du programme linéaire en écriture **mathématique** ainsi que l'interprétation en **français** de chacune des variables, équations
- La présentation de l'ensemble de votre programme : comment l'avez vous structuré ?
- La présentation des résultats (*valeur de la fonction objectif, temps d'exécution solveur, figure(s), tableau(x) ...*)
- L'interprétation des données et des résultats
- Si quelque chose n'a pas fonctionné, émettre des hypothèses

Nous ne souhaitons **pas de code** dans le rapport.

Le **maximum de page est fixé à 10**. Toutefois, ce nombre n'est pas une borne à atteindre.

Recapitulatif structure du projet

```
# Le zip ou tar.gz à rendre du dossier projet
|_ projet_R0_NOM1_PRENOM1_NOM2_PRENOM2.zip
   |_ truck_pulp.py
      |_ output_files/
         |_ *
      |_ *.py
   |_ projet_R0_NOM1_PRENOM1_NOM2_PRENOM2.pdf

# Le programme de test en dehors du dossier du projet
|_ test.py

# Le dossier des données en dehors du dossier du projet
|_ data/
   |_ *.data
```