SAÉ 2.04 BDD | Rapport

Table of Contents

1. Consignes du projet . . . 1
2. Version 1 . . . 2
   2.1. Présentation d'un exemple . . . 2
   2.2. Modèle pour l'exemple . . . 3
   2.3. Modélisation pour la Version 1 dans le cas général . . . 4
3. Version 2 : multimodalité et prise en compte des correspondances . . . 5
   3.1. Présentation d'un exemple . . . 5
   3.2. Modèle pour l'exemple . . . 6
   3.3. Modélisation pour la Version 2 dans le cas général . . . 6
   3.4. Implémentation de la Version 2 . . . 7
4. Fin du rapport . . . 7
   4.1. Barème sur 30 pts . . . 7

[Logo] | *logo.png*

Projet réalisé part FANTUZ CHROBOT Léo, LYOEN Quentin et GODDEFROY Arthur, groupe A.

1. Consignes du projet

Votre rapport doit suivre le plan donné dans ce document.

Contraintes à respecter pour le rapport

- [ ] format: Markdown qu'on peut lire sur gitlab, ou pdf, ou html - [ ] rapport dans un répertoire `graphes` *à la racine du dépôt git - [ ] rapport prêt le 21/06/2024; aucun délai supplémentaire ne sera accordé quelle que soit la raison donnée. Concrètement, on va récupérer la dernier daté au plus tard le 21/06/2024 et on ne verra même pas de version ultérieure du rapport, si elles existent. Minuit et une minute du 22/06/2024 sera trop tard - [ ] respecte le plan donné ci-dessous - [ ] garder les explications* ***en italique*** *jusque la fin pour s'y référer en écrivant le rapport - [ ] supprimer les explications* ***en italique*** *juste avant de rendre la version finale du rapport - [ ] le rapport est un texte* ***rédigé*** *avec des phrases intelligibles (on ne se contente pas de répondre laconiquement aux questions posées)*

Idéalement, le rapport est rédigé au fur et à mesure avec le calendrier donné dans le sujet:

- section Version 1 faite avant le 18/05/2024 (1pt/20 si c'est le cas) - section Version 2 faite avant le 08/06/2024 (1pt/20 si c'est le cas)

Finalement, l'utilisation d'un outils de génération de langage est autorisées, à condition de le faire intelligemment. En particulier, veillez à: - avoir un rapport cohérent avec un style cohérent sur la totalité du

document (niveau de langage, richesse du vocabulaire, termes utilisés, verbosité, ...) - un rapport trop verbeux est fastidieux à lire. Si vous utilisez un outil pour faire du texte verbeux inutile, on utilisera un outil pour en faire un résumé et on corrigera uniquement le résumé - les outils de génération insèrent parfois des phrases ne faisant pas partie du texte, mais qui s'adressent à l'interlocuteur (par exemple, pour vous informer que la limite de 2000 tokens est atteinte). La présence de telles phrases dans le rapport indique que vous n'avez pas relu et sera lourdement pénalisée.

Début du rapport *Tout ce qui précède sera enlevé pour la version finale*

2. Version 1

NOTE	Les parties situées dans un encadré correspondent aux réponses de notre rapport (le reste étant uniquement les consignes en italique).

```
SAE S2.02 -- Rapport pour la ressource Graphes
===

Noms des auteurs, groupe:
Arthur Goddefroy
Leo Fantuz Chrobot
Quentin Lyoen

Groupe: A
```

Version 1 : un seul moyen de transport

Cette section traite uniquement de la Version 1 du projet.

2.1. Présentation d'un exemple

Présenter un exemple concret de problème (données complètes pour la plateforme avec tous les moyens de transport, préférences de l'utilisatrices qui comprennent le moyen de transport choisi, le critère d'optimisation, et nombre d'itinéraires demandés). Donner la solution du problème du point de vue de l'utilisatrice, càd quels sont les itinéraires possibles, quels sont les meilleurs itinéraires et pourquoi. Pour l'instant on ne parle pas de graphes; on peut éventuellement faire des schémas.

```
Mr Nongaillard souhaite se rendre à l'IUT pour assurer un amphi de présentation de la sae
2.01/2.02. Il souhaite trouver le meilleur chemin pour s'y rendre par rapport à plusieurs
donnés : le temps du trajet en minute, le coût du trajet en euros et la quantité de CO2
émise. Par exemple, Mr Nongaillard souhaite se rendre à l'IUT le plus rapidement possible
tant que le prix ne dépasse pas 7 euros et ce, en train. Voici les chemins mis à
disposition par la SNCF:

trajet de la ville A à E: 1 euro, 10 minutes et 0.2 kg de CO2.
trajet de la ville E à Y: 6 euros, 5 minutes et 0.7 kg de CO2.
```

```
trajet de la ville Y à Z: 1 euro, 3 minutes et 0.9 kg de CO2.

trajet de la ville A à R: 3 euros, 15 minutes et 0.8 kg de CO2.
trajet de la ville R à Z: 2 euros, 7 minutes et 0.2 kg de CO2.

trajet de la ville A à T: 5 euros, 20 minutes et 0.5 kg de CO2.
trajet de la ville T à Z: 4 euros, 1 minute et 0.1 kg de CO2.

Ainsi, le chemin le plus court pour arriver le plus vite possible à l'IUT sans dépasser un
budget de 7 euros pour le trajet serait le chemin A -> R -> Z pour une durée de 22 minutes,
un coût de 5 euros et un taux de pollution de 1 kg de CO2 (tout autre chemin étant des
chemins valant plus de 7 euros, donc exclus).
```

2.2. Modèle pour l’exemple

Donner le graphe modélisant l’exemple ci-dessus. Donner la solution du problème (càd les meilleurs itinéraires) en tant que chemins dans le graphe.

Graphe pour le chemin en terme de prix:

[Logo] | *itineraire_prix.png*

Le chemin le plus court étant ainsi: [Nongaillard,A,R,Z,IUT] avec un coût de 5.

Le deuxième chemin le plus court étant ainsi: [Nongaillard,A,E,Y,Z,IUT] avec un coût de 8.

Le troisème chemin le plus court étant ainsi: [Nongaillard,A,T,Z,IUT] avec un coût de 9.

Graphe pour le chemin en terme de temps:

[Logo] | *itineraire_temps.png*

Le chemin le plus court étant ainsi: [Nongaillard,A,E,Y,Z,IUT] avec un coût de 18.

Le deuxième chemin le plus court étant ainsi: [Nongaillard,A,T,Z,IUT] avec un coût de 21.

Le troisème chemin le plus court étant ainsi: [Nongaillard,A,R,Z,IUT] avec un coût de 22.

Graphe pour le chemin en terme de CO2:

[Logo] | *itineraire_co2.png*

Le chemin le plus court étant ainsi: [Nongaillard,A,T,Z,IUT] avec un coût de 0.6.

Le deuxième chemin le plus court étant ainsi: [Nongaillard,A,R,Z,IUT] avec un coût de 1.

Le troisème chemin le plus court étant ainsi: [Nongaillard,A,E,Y,Z,IUT] avec un coût de 1.8.

2.3. Modélisation pour la Version 1 dans le cas général

Expliquer de manière abstraite comment, étant donné un problème de recherche d'itinéraire (plateforme avec tous types de lignes, moyen de transport choisi, critère d'optimisation, nombre d'itinéraires demandés) on peut construire un graphe permettant de résoudre le problème de recherche d'itinéraire. C'est à dire: - quels sont les sommets du graphe par rapport aux données du problème, - quelles sont ses arêtes, par rapport aux données du problème, - comment sont définis les poids des arêtes, - quel algorithme sur les graphes permet de résoudre le problème d'itinéraire (nom de l'algorithme, arguments).

Utiliser un vocabulaire précis sur les graphes.

```
Mr Nongaillard souhaite passer le moins de temps possible sur la route et ne souhaite pas
changer de moyen de transport. Il gardera ainsi le même moyen de transport du début à la
fin. Mr Nongaillard habite dans la ville A et L'IUT se situe dans la ville Z.

Mr Nongaillard possède trois cartes des environ : une qui lui donne le temps des trajets,
une qui lui donne le coût des trajets et une qui lui donne l'émission de CO2 des Trajets.
Il n'utilise qu'une seule carte à la fois. Ainsi, chaque sommet représente une ville dans
laquelle Mr Nongaillard peut se rendre afin d'arriver à l'IUT. Les arêtes quant à elles
constituent les routes empruntable par Mr Nongaillard en fonction du type de transport
souhaité (Train, Avion, Bus, Vélo...).

De manière générale, bien que Mr Nongaillard ne peut utiliser qu'une seule carte à la fois,
il peut tout de même connaître toutes les informations (prix, temps et taux de pollution)
de son trajet. Ainsi, chaque chemin contient 3 poids : son prix en euros, son temps en
minutes et son taux de pollution en CO2.

Pour finir, il est bon de mentionner que l'algorithme sur les graphes permettant ainsi de
calculer les chemins les plus courts est l'algorithme de dijkstra, car aucune valeur ne
peut être négative dans nos graphes. Il s'agit ainsi du meilleur algorithme pour réaliser
ces calculs de chemins optimaux (Tout cela se fait évidemment sur un graphe appelé
multigraphe permettant l'ajout d'arêtes typées nous permettant ainsi dans notre exemple de
disposer de plusieurs arêtes différentes pour un même graphe en fonction du type de
modalité de transport de ces dernières).

------------------------------

=== Implémentation de la Version 1

*Écrire une classe de test qui reprend l'exemple, définit toutes les données de la
plateforme, construit le graphe et calcule la solution.*
*Votre classe peut utiliser des assertions (test unitaire) ou bien afficher la solution.*
*Donner ici le **nom complet de la classe**, **la date et l'identifiant du commit à
regarder** et un **lien vers la page de cette classe sur gitlab qui correspond au bon
commit***.
```

```
*On insiste sur l'importance de spécifier le commit. En effet, quand vous commencerez la
Version 2, le code utilisé pour le test de la Version 1 sera modifié. Il se peut que vous
n'ayez pas le temps de finaliser la Version 2 et vous retrouver avec un code qui ne marche
pas même pour la Version 1. C'est pourquoi il est important de rédiger le rapport au fur et
à mesure et de donner ici un lien vers la version de votre code qui marche pour la Version
1 du projet.*

--------------
Nom de la classe: exempleGraphe (située dans le dossier graphes à la racine de notre
dépôt).
identifiant du commit: 080af3655b200d94b4be24202d7ebcc524d39100
date du commit: 18 Mai 2024 à 13h58

lien vers la page gitlab du dossier graphe: https://gitlab.univ-lille.fr/sae2.01-
2.02/2024/A3/-/tree/main/graphes?ref_type=heads

lien vers la page gitlab du commit du fichier exempleGraphe.java: https://gitlab.univ-
lille.fr/sae2.01-2.02/2024/A3/-
/blob/080af3655b200d94b4be24202d7ebcc524d39100/graphes/exempleGraphe.java



--------------
```

3. Version 2 : multimodalité et prise en compte des correspondances

Cette section explique la solution pour la Version 2 du projet. Le problème posé par cette version est de prendre en compte les correspondances entre deux moyen de transport differents.

3.1. Présentation d'un exemple

Afin de resoudre ce probléme : nous avons mis au points une solution qui diverge de la version 1 : chaque ville est séparée en plusieurs sommet a raison de 1 sommet par moyen de transport dans la ville

Présenter un exemple concret (plateforme, couts de correspondance, critère d'optimalité).

Mr Nongaillard souhaite se rendre à l'IUT pour assurer un amphi de présentation de la sae 2.01/2.02. Il souhaite trouver le meilleur chemin pour s'y rendre par rapport à plusieurs donnés : le temps du trajet en minute, le coût du trajet en euros et la quantité de CO2 émise. Par exemple, Mr Nongaillard souhaite se rendre à l'IUT le plus rapidement possible peux importe le moyen de transport. On partira du principe pour cet exemple qu'une correspondance coute 5 euros, prends 10 minutes et n'emmet aucun CO2 car Mr Nongaillard se deplace en trotinette.

```
trajet en train de la ville A à E: 1 euro, 10 minutes et 0.2 kg de CO2.
trajet en train de la ville E à Y: 6 euros, 5 minutes et 0.7 kg de CO2.
trajet en train de la ville Y à Z: 1 euro, 3 minutes et 0.9 kg de CO2.
```

```
trajet en train de la ville A à R: 3 euros, 15 minutes et 0.8 kg de CO2.
trajet en train de la ville R à Z: 2 euros, 7 minutes et 0.2 kg de CO2.
trajet en train de la ville A à T: 5 euros, 20 minutes et 0.5 kg de CO2.
trajet en train de la ville T à Z: 4 euros, 13 minute et 0.1 kg de CO2.
trajet en avion de la ville A à Y: 15 euros, 3 minutes et 1.5 kg de CO2
```

Donner la solution du problème du point de vue de l'utilisatrice (quels sont les itinéraires possibles, lesquels sont optimaux et pourquoi).

```
Du point de vue de l'utilisateur le meilleur chemin est :  Nongaillard, A_avion, Y_avion,
Y_train, Z_train,IUT qui ne lui prend que 13 minutes
```

Il est possible d'utiliser le même exemple que pour la Version 1 ou le modifier si pertinent.

L'exemple de la version restait pertinant à condition d'être legerement adptés c'est donc ce que l'on a fait.

3.2. Modèle pour l'exemple

Donner le graphe modélisant l'exemple ci-dessus. Donner la solution du problème (càd les meilleurs itinéraires) en tant que chemins dans le graphe.

3.3. Modélisation pour la Version 2 dans le cas général

Mêmes questions que pour la section correspondante de la Version 1, mais cette fois-ci les données d'entrée contiennent aussi des couts de correspondance. Vous pouvez expliquer l'entièreté de la solution pour la Version 2, ou bien indiquer clairement les différences par rapport à la solution proposée pour la Version 1.

Mr Nongaillard souhaite passer le moins de temps possible sur la route et ne souhaite pas changer de moyen de transport. Il gardera ainsi le même moyen de transport du début à la fin. Mr Nongaillard habite dans la ville A et L'IUT se situe dans la ville Z.

Mr Nongaillard possède trois cartes des environ : une qui lui donne le temps des trajets, une qui lui donne le coût des trajets et une qui lui donne l'émission de CO2 des Trajets. Il n'utilise qu'une seule carte à la fois. Ainsi, chaque sommet représente un couple ville_moyen de transport que Mr Nongaillard emprunte afin d'arriver à l'IUT. Les arêtes quant à elles constituent les routes empruntable par Mr Nongaillard en fonction du type de transport souhaité (Train, Avion, Bus, Vélo...) ou bien representent les correspondences entre deux moyens de transport quand elles relient deux sommets de la même ville.

De manière générale, bien que Mr Nongaillard ne peut utiliser qu'une seule carte à la fois, il peut tout de même connaître toutes les informations (prix, temps et taux de pollution) de son trajet. Ainsi, chaque chemin contient 3 poids : son prix en euros, son temps en minutes et son taux de pollution en CO2.

Pour finir, il est bon de mentionner que l'algorithme sur les graphes permettant ainsi de calculer les chemins les plus courts est l'algorithme de dijkstra, car aucune valeur ne peut être négative dans nos graphes. Il s'agit ainsi du meilleur algorithme pour réaliser ces calculs de chemins optimaux (Tout cela se fait évidemment sur un graphe appelé multigraphe permettant l'ajout d'arêtes typées nous permettant

ainsi dans notre exemple de disposer de plusieurs arêtes différentes pour un même graphe en fonction du type de modalité de transport de ces dernières).

3.4. Implémentation de la Version 2

Écrire une classe de test qui reprend l'exemple, définit toutes les données de la plateforme, construit le graphe et calcule la solution. Votre classe peut utiliser des assertions (test unitaire) ou bien afficher la solution. Donner ici le nom complet de la classe, la date et l'identifiant du commit à regarder et un **lien vers la page de cette classe sur gitlab qui correspond au bon commit. En particulier, il peut s'agir de la même classe que celle donnée pour la Version 1, mais un commit différent.

Version 3 : optimisation multi-critères ---

Suivre le même plan que pour les deux autres sections. Pour l'exemple, veillez à spécifier toutes les données des problèmes. En particulier, on ajoute ici l'expression des préférences d'optimisation de l'utilisatrice. Comme précédemment, il est possible d'utiliser le même exemple et simplement l'enrichir.

4. Fin du rapport

4.1. Barème sur 30 pts

Toute question sur le barème est à adresser à iovka.boneva@univ-lille.fr

- Rapport non rendu à temps → note 0
- **(7, décomposé comme suit)** Divers
- **(1,5)** Respect de la structure du rapport
- **(1,5)** Section Version 1 rendue pour le 18/05/2024. Cette version peut contenir les parties en italique.
- **(1,5)** Section Version 2 rendue pour le 08/06/2024. Cette version peut contenir les parties en italique.
- **(1)** Utilisation de vocabulaire précis sur les graphes (termes vu en cours, noms des algorithmes, etc.)
- **(1,5)** Style d'écriture fluide et compréhensible
- **(8, décomposé comme suit)** Solution pour la Version 1
- **(2)** Exemple pertinent (illustre tous les aspects du problème) et lisible (en particulier, ni trop grand ni trop petit, bien présenté)
- **(4)** Le modèle de l'exemple permet de trouver la solution sur l'exemple. La modélisation pour le cas général permet de résoudre le problème posé
- **(2)** L'implémentation de l'exemple est correcte et fonctionnelle
- **(6, décomposé comme suit)** Solution pour la Version 2
- **(1)** Exemple pertinent
- **(4)** le modèle de l'exemple permet de trouver la solution sur l'exemple. La modélisation pour le cas général permet de résoudre le problème posé

- **(1)** L’implémentation de l’exemple est correcte et fonctionnelle
- **(3)** Qualité de la description de la solution (concerne les sections "Modèlisation dans le cas général" pour les Versions 1 et 2):
- La modélisation pour le cas général est décrite de manière abstraite mais précise et complète. Pour vous donner une idée, un·e étudiant·e de BUT qui a validé les ressources Graphes et Dev devrait être en mesure d’implémenter votre solution d’après la description que vous en faites, sans avoir à trop réfléchir.
- **(6)** Solution pour la Version 3: mêmes critères que pour la Version 2