

A la conquête d'Hollywood

Informatique - IUT d'Orléans - Année 2021-2022

1 Déroulement

Le travail de la *SAE 2.02 Exploration Algorithmique d'un problème* sera effectué pendant la période allant de la semaine du 8 avril 2024 à la semaine du 3 juin 2024. Il y aura 8 séances de 1h30, soit 12h au total, de travail en présentiel.

2 Présentation du sujet

Un jeune cinéphile passionné de sport souhaite se faire un nom à Hollywood en devenant le coach sportif incontournable des stars. Pour cela il souhaite adopter une stratégie simple: se faire embaucher par les acteurs ou actrices "qui comptent" puis attendre que ces personnalités centrales d'Hollywood lui fassent bonne presse. Il espère ainsi accroître rapidement sa notoriété.

Pour que les choses aillent plus vite, il souhaite donc trouver les bons candidats afin d'optimiser ses chances de succès. Il décide d'en parler à un ami mathématicien férus de cinéma qui lui parle du jeu des **six degrés de Kevin Bacon**. Après quelques explications (et un bon mal de crâne) notre agent est convaincu: il tient là le concept qui lui ouvrira toutes les portes de l'industrie du cinéma!

3 Le nombre de bacon et la théorie des graphes

Présentons brièvement les explications de l'ami mathématicien.

L'idée principale est de représenter les collaborations des acteurs et actrices d'Hollywood par un graphe, où les sommets seraient les acteurs/actrices et où il y a une arête entre deux sommets si les acteurs/actrices correspondant.e.s ont tourné dans un même film. Appelons ce graphe le **graphe des collaborations** et notons-le G_c par la suite.

Le nombre de Bacon d'un acteur ou une actrice permet de déterminer son éloignement dans le graphe G_c par rapport à Kevin Bacon. Plus formellement, le nombre de Bacon d'un acteur v correspond à la distance (c'est-à-dire le nombre d'arête d'un plus court chemin) entre v et le sommet représentant Kevin Bacon dans G_c .

Ainsi, Kevin Bacon a un nombre de Bacon de 0, un acteur ayant tourné avec Kevin Bacon a un nombre de Bacon de 1, un acteur ayant tourné avec un acteur ayant lui-même tourné avec Kevin Bacon aura un nombre de Bacon de 2, etc

Les 6 degrés de Kévin Bacon est un jeu qui est apparu dans les années 90 et qui consiste à chercher le nombre de Bacon d'un acteur. Ce concept est proche des notions de 6 degrés de séparation et de graphes petit monde. Le site internet The Oracle of Bacon fournit une interface permettant d'interroger une base de données fournissant une réponse à cette question.

4 Objectifs de la SAé

En reprenant l'idée du site The Oracle Of Bacon, nous allons implémenter un programme permettant de répondre à un certain nombre de requêtes permettant d'évaluer la centralité des acteurs d'Hollywood. Cet outil sera sans aucun doute d'une aide précieuse pour notre future coach!



Figure 1: Walk of Fame (wolfgang.mller54/Wikimedia Commons/CC-BY-2.0)

Le programme sera codé en Python et devra fournir différentes fonctions permettant de répondre aux requêtes décrites dans la section 6. Vous utiliserez la bibliothèque Networkx afin de représenter le graphe G_c .

Le jeu de données sur lequel vous travaillerez a été récupéré via le site The Oracle of Bacon et est issu des données de Wikipédia. De ce fait il hérite de la licence CC-BY-SA associée au contenu de Wikipedia.

5 Modalités d'évaluation

Le projet sera à réaliser en binome, dont les deux membres doivent appartenir au même groupe de TD. Vous devrez remplir le test «Composition des binômes» idéalement à la fin de votre première séance et au plus tard **avant le 14 avril minuit**, en indiquant l'identité de votre partenaire de binome (nom, prénom). Une seule réponse par groupe suffit.

Le rendu devra comporter, sous la forme d'une archive au format zip:

- un dépôt git avec vos fichiers python implémentant l'oracle de Bacon et les différentes requêtes demandées,
- un rapport au format pdf décrivant les fonctionnalités implémentées ainsi que les problèmes rencontrés et les possibilités d'amélioration de votre solution. Votre rapport décrira également les tâches qui auront été réalisées par chaque membre du binome. Pour finir, vous y décrirez les évaluations expérimentales réalisées (voir la section 7 pour plus de détails).

Le rendu se fera sur Celene en utilisant la zone de dépôt prévue à cet effet. La date limite de dépôt du code sera précisée ultérieurement.

6 Requêtes à implémenter

Les requêtes suivantes seront à implémenter et intégrer à votre application. Différentes questions sont également posées au fil des énoncés. Vous y répondrez et reporterez vos réponses dans votre rapport.

6.1 Échauffement

La première étape de votre travail sera de proposer une fonction permettant de convertir le jeu de données (au format json) représentant les collaborations en un graphe Networkx exploitable.

6.2 Collaborateurs en communs

Cette requête consiste à renvoyer, pour deux acteurs/actrices donné.e.s, l'ensemble des acteurs/actrices qui ont collaboré.e.s avec ces deux personnes.

Comment exprimeriez-vous cette notion (ensemble des collaborateurs en commun) en terme de théorie des graphes? Pouvez-vous donner une borne inférieure sur le temps nécessaire à l'exécution de votre fonction?

6.3 Collaborateurs proches

Nous vous fournissons une fonction prenant en paramètre un acteur et un entier k et renvoyant la liste des acteurs qui se trouvent à distance au plus k de l'acteur en question. Étudiez ce code.

Reconnaissez-vous l'algorithme classique en théorie des graphes qui est au coeur de ce programme?

Grâce à la fonction précédente, comment pouvez-vous déterminer si un acteur se trouve à distance k d'un autre acteur?

Tentons maintenant de déterminer la distance entre deux acteurs. Est-ce que ré-utiliser la fonction précédente vous semble intéressant? Donnez la complexité (asymptotique) d'un tel algorithme.

Comment pouvez-vous maintenant modifier la fonction qui vous a été fourni afin de trouver la distance entre deux acteurs? Donnez la complexité d'un tel algorithme.

6.4 Qui est au centre d'Hollywood?

On cherche maintenant à déterminer la centralité d'un acteur. Pour cela, on veut déterminer la plus grande distance qui le sépare d'un autre acteur dans le graphe.

Quelle notion de théorie des graphes permet de modéliser cela? Proposez une fonction qui calcule la centralité d'un acteur dans G_c .

A l'aide de la fonction précédente, écrivez une autre fonction qui va déterminer l'acteur le plus central du graphe G_c . Plus formellement, notons $c(G, v)$ la centralité d'un acteur v . Dans ce cas, le centre du graphe est un sommet s tel que $c(G, s) = \min_{v \in V(G)} c(G, v)$.

6.5 Une petite famille

Nous allons maintenant tenter de déterminer si deux acteurs peuvent être très éloignés dans G_c . Plus précisément, vous fournirez une fonction permettant de déterminer la distance maximum dans G_c entre toute paire d'acteurs/actrices. Est-ce que ce nombre est bien inférieur ou égal à 6 pour le jeu de données fourni?

6.6 Bonus

- Proposez une méthode similaire à celle calculant le centre du graphe mais qui se restreint ici à déterminer le centre d'un groupe d'acteur (On remarquera qu'un tel sommet ne fait pas nécessairement parti du groupe en question).
- Faites en sorte, quand cela a du sens, que vos fonctions renvoient un sous-graphe de G_c plutôt qu'une simple liste d'acteur. Par exemple, proposer une variante de la fonction déterminant les collaborateurs proches d'un sommet v qui renverra le sous-graphe induit par v et par tous les sommets à distance k de v .

7 Un peu d'efficacité

L'efficacité de vos méthodes, tant sur le temps d'exécution que sur la mémoire utilisée, est bien entendu essentielle. Par exemple, calculer la distance entre 2 sommets peut se faire de différentes manières, comme évoqué précédemment, impliquant des temps de calcul différents. Il est également possible de pré-calculer un certain nombre de distance et d'utiliser ces informations pour calculer des distances entre des paires de sommets plus éloignés.

Pour cette dernière partie du travail, vous vous attacherez donc à proposer différentes implémentations du calcul de distance et à évaluer ces différentes propositions. La description de vos évaluations expérimentales ainsi que vos conclusions devront apparaître dans votre rapport.

8 Le rendu

La date limite de rendu est fixée au **vendredi 31 mai à 23h59**. Un dépôt sera disponible à cette fin sur la page Celene dédiée aux SAé.

Pour rappel, le rendu devra contenir:

- le **code Python** de votre application
- le **rappor**t détaillant le travail réalisé, la répartition des tâches en les membres du binomes, les réponses aux questions ainsi qu'une évaluation expérimentale (mesure de performance + analyse) des différentes solutions mises en place au cours de la SAé.

Il vous est également demandé de fournir un **document individuel** d'une page maximum, vous incitant à prendre du recul sur votre travail (voir détails plus bas).

Pour les questions sur la complexité de vos algorithmes, sauf mention du contraire dans la consigne, il vous est demandé de fournir la complexité asymptotique du temps d'exécution dans le pire cas (borne supérieure).

8.1 Détails sur le rendu individuel

Dans le document individuel (1 document par étudiant), vous détaillerez:

- le travail que vous avez personnellement réalisé ainsi que les difficultés rencontrées
- les acquis de votre formation que vous avez été amené à mobiliser au cours de cette SAé ainsi que les nouvelles notions apprises au cours de cette situation d'apprentissage.
- la démonstration de l'acquisition des compétences liées à cette SAé.

Vous trouverez ci-dessous la liste des apprentissages critiques ainsi que des ressources liées à cette SAé.

8.1.1 Apprentissages critiques

- AC12.01 : Analyser un problème avec méthode (découpage en éléments algorithmiques simples, structure de données...)
- AC12.02 : Comparer des algorithmes pour des problèmes classiques (tris simples, recherche...)
- AC12.03 : Formaliser et mettre en œuvre des outils mathématiques pour l'informatique

8.1.2 Ressources mobilisées et combinées

- R2.01 : Développement orienté objets
- R2.02 : Développement d'applications avec IHM
- R2.03 : Qualité de développement
- R2.07 : Graphes
- R2.09 : Méthodes numériques

8.2 Détails sur le code

Votre application devra à minima contenir deux fichiers:

1. le fichier principal nommé `oracle.py` permettant le lancement de l'application
2. un fichier nommé `requetes.py` contenant les différentes fonctions à implémenter.

Les fonctions du fichier `requetes.py` devront respecter l'API suivante:

```
# Q1
def json_vers_nx(chemin):

# Q2
def collaborateurs_comuns(G,u,v):

# Q3
def collaborateurs_proches(G,u,k):
def est_proche(G,u,v,k=1):
def distance_naive(G,u,v):
def distance(G,u,v):

# Q4
def centralite(G,u):
def centre_hollywood(G):

# Q5
```

```
def eloignement_max(G:nx.Graph):  
  
    # Bonus  
def centralite_groupe(G,S):
```

Une attention particulière sera portée à la qualité de votre code (lisibilité, commentaires, test,...)

9 Références utiles

- Le site de l'Oracle de Bacon: <https://oracleofbacon.org/center.php>
- Le nombre d'Erdos qui est un concept similaire appliqué aux collaborations mathématiques: <https://oakland.edu/enp/trivia/>
- Un article résumant le concept du nombre de Bacon ainsi que son apparition: <https://blogs.ams.org/mathgradblog/2013/11/22/degrees-kevin-bacon/>