

APPERO

Optimisation Hivernale

Jose A. Henriquez Roa

Contents

1	Contributions	1
1.1	Algorithm principal: implémentation	1
1.2	Algorithm principal: présentation	1
1.3	Hungarian algorithm: implémentation	1
1.4	Hungarian algorithm: présentation	1
1.5	Floyd-Warshall algorithm: implémentation	1
1.6	Hierholzer algorithm: implémentation	2
1.7	Test suit	2
2	Éléments appris	2
2.1	Chinese postman problem	2
2.2	Hungarian Algorithm	2
3	Sources principales	2

1 Contributions

Dans l'espoir de faire le projet par moi-même, j'ai choisi un groupe vide le 8 août. J'ai commencé à faire des recherches pour résoudre le problème le même jour. Au moment où l'annonce a été faite que les groupes de moins de 3 ou 4 membres ont été fusionnés le 17 août, j'avais déjà une implémentation fonctionnelle d'un algorithme pour résoudre le problème.

Lors de notre première réunion de groupe, la décision a ensuite été prise de répartir l'implémentation et la présentation des trois principaux algorithmes que j'ai utilisés entre nous trois. Leurs nouvelles implémentations devaient ensuite être intégrées ultérieurement. Cependant, au fil du temps, la décision a été prise de garder mon implémentation intégrée dans l'algorithme principal. C'est la raison pour laquelle ils sont tous répertoriés ici.

1.1 Algorithm principal: implémentation

Path: `snowymontreal/solver.py`

C'est l'algorithme qui présente la solution aux principaux Problème: **Route Inspection Problem** ou **Chinese Postman Problem**.

1.2 Algorithm principal: présentation

Path: `doc/presentation.mp4`

J'étais responsable de la présentation de celui-ci. Dans la vidéo finale, celle-ci se trouve dans la section *Introduction*.

1.3 Hungarian algorithm: implémentation

Path: `snowymontreal/hungarian.py`

Cet algorithme est utilisé pour trouver la couplage minimale dans le graphe bipartite construit à partir des sommet de degré plus grand et plus petit que 0 (plus d'informations à ce sujet sur la vidéo de présentation). Cet algorithme a été développé et publié par Harold Kuhn en 1955.

1.4 Hungarian algorithm: présentation

Path: `doc/presentation.mp4`

J'étais également responsable de la présentation de cet algorithme. Celui-ci se trouve dans la section respective *Hungarian Algorithm* de la vidéo finale.

1.5 Floyd-Warshall algorithm: implémentation

Path: `snowymontreal/floyd-warshall.py`

Cet algorithme est utilisé pour trouver le chemin de poids minimal entre tous les sommet du graph. Il est principalement utilisé pour la section de duplication de chemin de l'algorithme principal (plus d'informations à ce sujet sur la vidéo de présentation). Cet algorithme a été développé et publié par Rober Floyd et Stephen Warshall en 1962.

1.6 Hierholzer algorithm: implémentation

Path: snowymontreal/hierholzer.py

Après duplication de chemin, cet algorithme est utilisé pour trouver le chemin eulérien final retourné (plus d'informations à ce sujet sur la vidéo de présentation). Cet algorithme a été développé et publié par Carl Hierholzer en 1873. Après duplication de chemin, cet algorithme est utilisé pour trouver le chemin eulérien final retourné (plus d'informations à ce sujet sur la vidéo de présentation). Cet algorithme a été développé et publié par Carl Hierholzer en 1873.

1.7 Test suit

Path: test/test.py

Il s'agit d'une simple "black box testsuit" pour vérifier la sortie de l'algorithme principal.

2 Éléments appris

2.1 Chinese postman problem

J'ai appris un moyen de résoudre le "Chinese postman problem".

2.2 Hungarian Algorithm

Je me suis également familiarisé avec le fonctionnement interne du "Hungarian Algorithm".

3 Sources principales

- <https://blog.scienceborealis.ca/the-chinese-postman-and-the-trick-or-treater>
- https://www-m9.ma.tum.de/graph-algorithms/directed-chinese-postman/index_en.html
- <https://www.geeksforgeeks.org/maximum-bipartite-matching>
- <https://www.geeksforgeeks.org/euler-circuit-directed-graph>