ISEN | école d'ingénieurs

ALL IS DIGITAL!

CIR3 Brest – Projet théorie des graphes

4 novembre 2024

1 Modalités du projet

Le projet se réalisera en binôme ou trinômes, devra être rendu, avec un code source et un rapport sur Moodle. Le rapport n'a pas besoin d'être étendu, mais il doit présenter la part de travail du binôme, les réponses aux questions ci-dessous et les résultats obtenus, et éventuellement les choix faits lors de l'élaboration du programme.

Un programme python (`HexGridViewer`) vous est fourni afin de gérer l'affichage, disponible sur Moodle. Seuls des outils d'affichage sont présents (sauf la méthode , qui vous sera utile...). Le fichier est commenté et documenté. Vous pouvez lancer ce programme, son *main* permet de visualiser les capacités du programme.

Des soutenances auront lieu après le dépôt du rapport, lors du dernier cours (les dates seront indiquées sur Moodle).

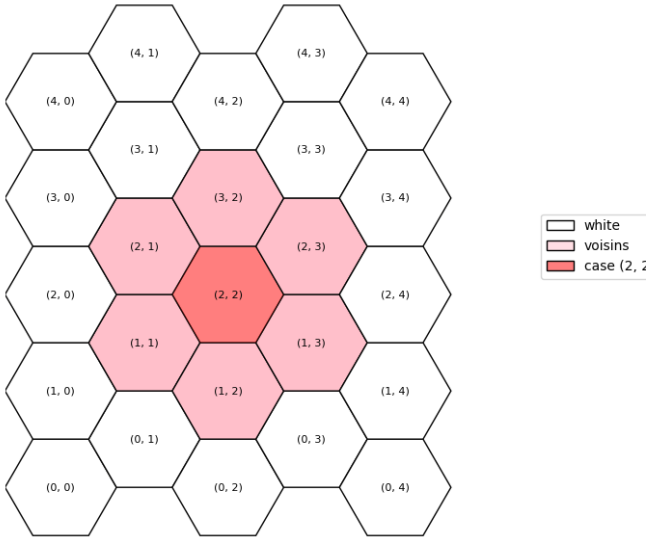
— Cette soutenance a pour but de vérifier l'implication de chacun et que tout le monde maîtrise le contenu du projet et que le programme est bien de vous.

Le code source rendu, devra présenter le résultat des questions ci-dessous, utilisant **la structure de données développée lors des TPs**. Ce sera l'occasion pour le binôme de manipuler leurs propres structures et les comparer si elles diffèrent. Ce code devra être **clair, structuré et commenté.**.

— Dans votre programme, vous pouvez mettre autant de fichier que souhaité. Il devra comporter un *main* clair qui permet de lancer et de répondre aux questions du sujet, une à une.

2 Présentation

Le but du devoir est d'appliquer différents algorithmes vus lors du cours de théorie des graphes et de proposer des solutions à des problèmes "concrets". Ici, nous modéliserons une zone géographique, représentée en 2D, grâce à des cases hexagonales. Les grilles hexagonales, composées de cases hexagonales, sont particulières. En effet, chaque case possède 6 voisins (sauf sur les bordures). Ici, par exemple, la case en coordonnées marquées comme (2, 2) (en rouge) a 6 voisins (en rose).



Pour simplifier la gestion de ce type de grille, le programme python mis à votre disposition permet d'initialiser de créer un objet avec de dimensions et , et étant donné les coordonnées d'une case (tuple x et y), récupérer les coordonnées des cases voisines (liste de tuples x et y). Ici dans l'exemple ci-dessus, les voisins de la case aux coordonnées $(2, 2)$ sont $(1, 1), (2, 1), (3, 2), (2, 3), (1, 3)$ et $(1, 2)$.

À noter qu'il n'y a pas d'unique bonne réponse aux questions. En fonction de vos choix, vous pouvez proposer des solutions uniques et interéssantes. Votre rapport et la soutenance seront là pour discuter des réponses que vous apporterez aux questions posées.

3 Questions

Questions : modélisation de base

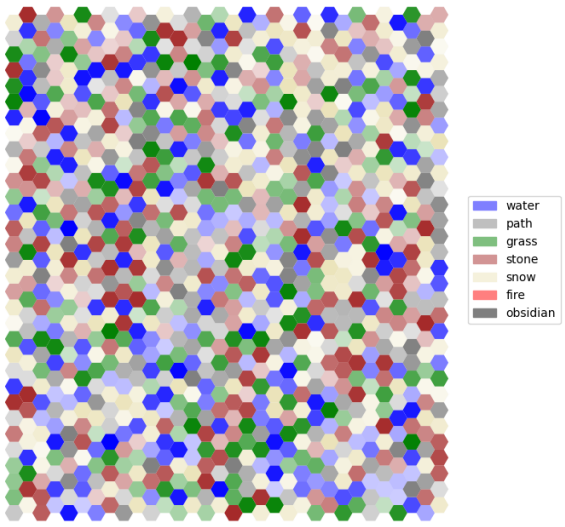
Q.1 Proposez une implémentation d'un graphe, qui représente une grille hexagonale et qui possède toutes les propriétés d'un graphe.

Q.2 Proposez une extension de cette implémentation, permettant :

- de labeliser les sommets par un type de terrain de votre choix (herbe, montagne, route, eau, etc...) ;
- de labeliser les sommets par une altitude.

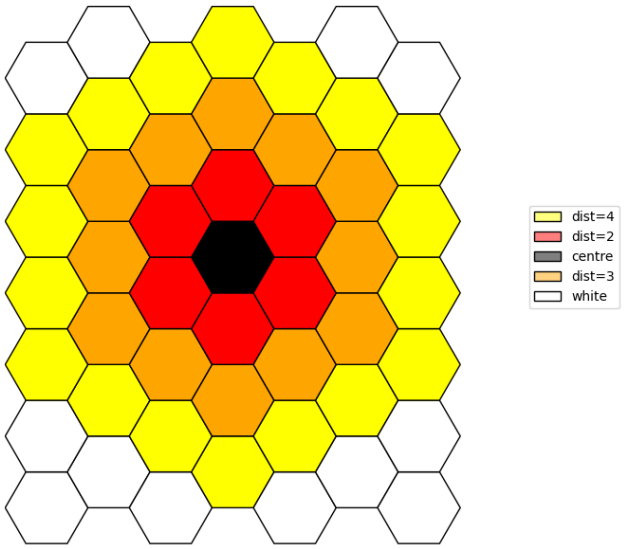
Q.3 Tester ce programme en ajoutant des types de terrain aléatoire et des altitudes aléatoires. Vous pouvez afficher le résultat grâce au programme *hexgrid_viewer.py*, en utilisant les couleurs pour signifier les terrains et la transparence pour signifier l'altitude.

Voici un exemple de rendu :



Questions : algorithmes de génération

Q.4 Quel algorithme utiliser pour générer une zone régulière qui s'étend sur la carte (i.e. toutes les cases à distance i d'une case) et comment l'adapter ? Implémentez-le.

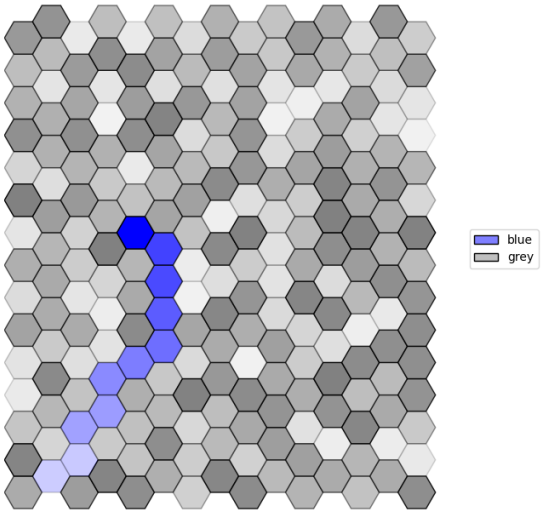


Exemple des cases à distance i de la case noire

Q.5

(a) créer une méthode permettant de trouver le sommet le plus haut de votre carte.

(b) Quel algorithme permettrait de tracer des rivières à partir d'un point donné sur la carte, en ajoutant une contrainte d'altitude descendante en prenant le chemin le plus long possible ?

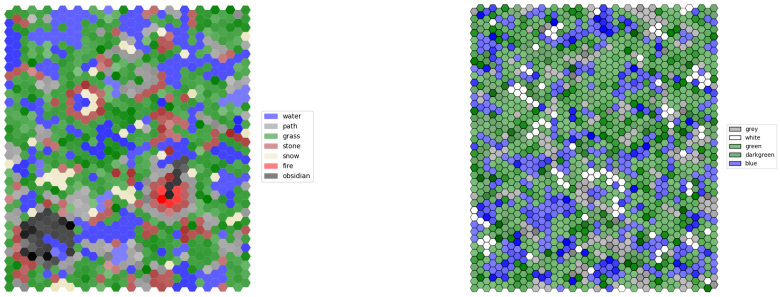


Exemple de "rivière" générée.

(c) Que pouvez-vous ajoutez pour créer des embranchements de rivières ? Quelle est cette structure obtenue ?

Q.6 Proposez maintenant un algorithme, qui, s'inspirant des deux prédécedents, génère une carte aléatoirement, de sorte à ce que les altitudes soient "logiques" et que les types de terrains aient une cohérence, avec des rivières.

Extension bonus : l'eau peut ne pas être une rivière, par exemple, avec les lacs. Quelle contrainte cela ajoute au programme ? Comment faire ?



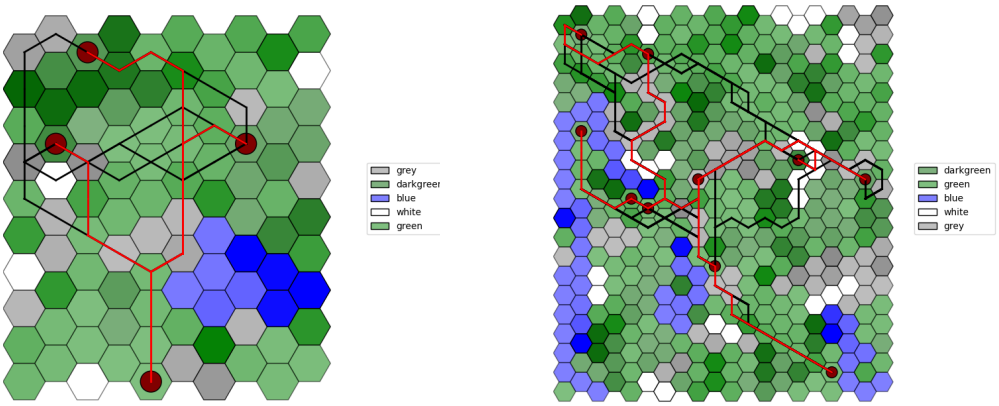
Exemples de cartes générées.

Questions : exploitation des données

Q.7 Posez des villes aléatoirement sur votre carte. Quel algorithme utiliseriez-vous pour générer les routes entre ces villes de sorte à ce qu'elles soient le plus rapide à emprunter (sans les contraintes de terrain et d'altitude) ? Implémentez votre solution. Quelle est la complexité de votre algorithme ? Comparez les temps de calculs de cet algorithme en fonction de la taille de votre grille hexagonale et du nombre de villes souhaité, grâce à un graphique.

Q.8 Quelle modification apporter afin de pouvoir prendre en compte les différences de type et d'altitude de vos cases ? Comment faire pour qu'il soit impossible de traverser de l'eau ? Comparez les résultats avec et sans ces contraintes.

Q.9 Quel algorithme utiliser pour créer un réseau de routes le moins couteux possible entre x villes, pour qu'elles sont toutes accessibles les unes par rapport aux autres ?



Exemples de routes générées. Ici, les routes noires sont elles qui interconnectent toutes les villes, et les routes rouges est le réseau de route minimal.
Ici, on considère les trajets impossibles dans l'eau et certaines parties du terrain moins praticable. Ici, deux chemins $A \to B$ et $B \to A$ peuvent être différents, en fonction de la topologie.

Q.10 Créer un algorithme, qui permet à un marchand de visiter toutes les villes en minimisant son parcours sur la route et de revenir à sa ville initiale.

Extension bonus : Admettons que nous disposons des moyens de transports autres que terrestres. Quelles modifications apporteriez-vous afin d'ajouter des bateaux ou des avions, par exemple ? (Implémentation non demandée, juste des pistes de réflexion).