Rapport de Projet d'Algorithmie

Table des matières

1. Introduction
2. Description du Projet
3. Structures de Données Utilisées
4. Méthodes Principales
5. Procédure de lancement
6. Conclusion

1. Introduction

Ce rapport présente le travail réalisé dans le cadre du projet d'algorithmie consistant à concevoir une application permettant d'établir la table de routage de chaque nœud d’un réseau de 100 nœuds. L'objectif principal est de décrire les différentes étapes de conception et d'implémentation, ainsi que les choix effectués en termes de structures de données et d'algorithmes.

2. Description du Projet

Le projet consiste à développer une application capable de générer un réseau de 100 nœuds, répartis en trois niveaux hiérarchiques (Tier 1, Tier 2 et Tier 3). Chaque nœud est connecté à un certain nombre de ses pairs, et l'application doit être capable d'établir la table de routage pour chaque nœud, permettant ainsi de déterminer le chemin le plus court entre deux nœuds donnés.

3. Structures de Données Utilisées

Dans notre implémentation, nous avons utilisé deux principales structures de données :

- **Graphes**: Nous avons représenté le réseau sous forme de graphe, où chaque nœud est un sommet et chaque lien entre les nœuds est une arête. Nous avons utilisé la bibliothèque NetworkX pour la manipulation des graphes.
- **Listes et Dictionnaires**: Nous avons utilisé des listes et des dictionnaires Python pour stocker les informations sur les nœuds du réseau, leurs connexions et les tables de routage.

4. Méthodes Principales

4.1. Création du Réseau

La classe Reseau est responsable de la création du réseau en fonction des spécifications données. Les nœuds sont répartis en trois niveaux hiérarchiques et des liens sont établis entre eux selon des probabilités spécifiées.

4.2. Parcours en Largeur

Nous utilisons un parcours en largeur pour vérifier si tous les nœuds du réseau sont accessibles à partir d'un nœud de départ donné. Cela nous permet de vérifier la connexité du réseau.

4.3. Algorithme de Dijkstra

L'algorithme de Dijkstra est utilisé pour calculer les plus courts chemins à partir d'un nœud source vers tous les autres nœuds du réseau. Cela nous permet de construire les tables de routage pour chaque nœud.

4.4. Affichage du Réseau

Nous avons développé une classe ReseauGraphique qui hérite de la classe Reseau et qui permet d'afficher graphiquement le réseau à l'aide de la bibliothèque Matplotlib. Cette classe permet également de sélectionner deux nœuds et d'afficher le chemin le plus court les reliant.

5. Procédure de lancement

Pour lancer le code, suivez ces étapes simples :

- Assurez-vous d'avoir Python installé sur votre système et d'avoir les bibliothèques nécessaires installées. Dans ce cas, les bibliothèques requises sont networkx et matplotlib. Vous pouvez les installer en exécutant la commande suivante dans votre terminal ou votre invite de commandes :
    - pip install matplotlib
    - pip install networkx

- Une fois que Python et les bibliothèques sont installés, exécuter le code.

- Une fois lancé, le programme générera un réseau de nœuds et vérifiera sa connexité en affichant un message de confirmation dans le terminal. Par ailleurs, si le graphe généré n’est

pas connexe, le code affichera un message d’erreur dans le terminal et générera un autre graphe, ainsi de suite jusqu’à ce qu’il soit connexe.

- La prochaine étape est de fermer la page et un message dans le terminal vous demandera de rentrer des nœuds pour afficher les chemins les reliant. A la fin de chaque affichage d’un chemin, on vous demandera si vous voulez rentrer d’autres nœuds pour en connaître le chemin entre ou vous arrêtez, ce qui fermera la page du code et terminera son exécution. De plus, il est bon de rajouter qu’en cliquant sur un nœud du graphe, vous pourrez voir afficher dans le terminal tous les nœuds reliés à celui-ci, c’est une option supplémentaire du code.

- Assurez-vous de suivre les instructions qui s'affichent dans le terminal pour sélectionner les nœuds et observer les chemins. En respectant ces étapes, le code devrait fonctionner correctement et vous permettre d'explorer le réseau généré.

6. Conclusion

Ce projet nous a permis de mettre en œuvre différentes notions d'algorithmie, notamment la manipulation de graphes et l'utilisation d'algorithmes de recherche de chemins. L'application développée remplit les objectifs fixés, en permettant la création et la visualisation d'un réseau ainsi que le calcul des tables de routage pour chaque nœud.

Le code complet de l'application est fourni en annexe.