

# Rapport : Implémentation de l'Algorithme de Hopcroft-Karp avec Interface Graphique en Java

---

## 1. Introduction

Ce projet consiste en une implémentation de l'algorithme de **Hopcroft-Karp** pour trouver un **couplage maximum** dans un graphe biparti. L'algorithme est présenté dans une interface graphique interactive développée en Java à l'aide de la bibliothèque **Swing**. L'interface permet à l'utilisateur de saisir les données du graphe, de visualiser son exécution étape par étape, et d'afficher le résultat final.

---

## 2. Contexte et Objectifs

- **Contexte** : L'algorithme de Hopcroft-Karp est un algorithme efficace pour résoudre le problème du couplage maximum dans un graphe biparti. Il est largement utilisé en théorie des graphes et en optimisation combinatoire.
  - **Objectifs** :
    - Implémenter l'algorithme de Hopcroft-Karp en Java.
    - Créer une interface graphique pour visualiser le graphe et le déroulement de l'algorithme.
    - Permettre à l'utilisateur de saisir les données du graphe et d'observer les étapes de l'algorithme en temps réel.
- 

## 3. Structure du Code

Le code est structuré en deux parties principales :

### 1. Implémentation de l'algorithme de Hopcroft-Karp :

- **BFS (Breadth-First Search)** : Utilisé pour calculer les distances des sommets non appariés.
- **DFS (Depth-First Search)** : Utilisé pour trouver des chemins d'augmentation.
- **Initialisation du graphe** : Le graphe est représenté par des listes d'adjacence.
- **Calcul du couplage maximum** : L'algorithme itère entre BFS et DFS jusqu'à ce qu'aucun chemin d'augmentation ne soit trouvé.

### 2. Interface Graphique :

- **Saisie des données** : L'utilisateur peut entrer le nombre de sommets dans les ensembles U et V, ainsi que les arêtes du graphe.
  - **Visualisation du graphe** : Le graphe est dessiné dans un panneau dédié (`GraphPanel`), avec des sommets colorés et des arêtes représentées par des lignes.
  - **Affichage des logs** : Les étapes de l'algorithme sont affichées dans une zone de texte (`logArea`) pour permettre à l'utilisateur de suivre l'exécution.
  - **Bouton de calcul** : Un bouton permet de lancer l'algorithme et d'afficher le résultat.
-

## 4. Fonctionnalités Clés

### 1. Saisie des Données :

- L'utilisateur peut entrer :
  - Le nombre de sommets dans les ensembles U et V.
  - Les arêtes du graphe au format u - v, séparées par des virgules.

- Exemple d'entrée :

U = 3, V = 3

Arêtes : 1-1, 1-2, 2-2, 3-3

### 2. Visualisation du Graphe :

- Les sommets de U sont représentés en bleu, ceux de V en rouge.
- Les arêtes sont dessinées en noir.
- Les arêtes du couplage maximum sont mises en évidence en vert.

### 3. Déroulement de l'Algorithme :

- Les étapes de l'algorithme (BFS, DFS, couplages trouvés) sont affichées dans la zone de logs.
- Des pauses sont ajoutées entre les étapes pour permettre une visualisation en temps réel.

### 4. Résultat Final :

- La taille du couplage maximum est affichée dans une boîte de dialogue.
  - Le graphe est mis à jour pour montrer les arêtes du couplage maximum.
- 

## 5. Exemple d'Exécution

### 1. Entrées :

- U = 3, V = 3
- Arêtes : 1-1, 1-2, 2-2, 3-3

### 2. Déroulement :

- Initialisation du graphe.
- Exécution de BFS pour calculer les distances.
- Exécution de DFS pour trouver des chemins d'augmentation.
- Affichage des couplages trouvés (par exemple, U1-V1, U2-V2, U3-V3).

### 3. Résultat :

- Taille du couplage maximum : 3
  - Visualisation du graphe avec les arêtes du couplage en vert.
- 

## 6. Points Forts du Projet

- **Interface Utilisateur Intuitive** : L'interface graphique permet une interaction facile avec l'algorithme.

- **Visualisation en Temps Réel** : Les étapes de l'algorithme sont affichées en temps réel, ce qui facilite la compréhension.
  - **Gestion des Erreurs** : Des messages d'erreur sont affichés si les entrées sont incorrectes.
  - **Extensibilité** : Le code est modulaire et peut être étendu pour inclure d'autres fonctionnalités (par exemple, la visualisation de graphes plus complexes).
- 

## 7. Limites et Améliorations Possibles

- **Limites** :
    - Le code ne gère pas les graphes de très grande taille (plus de 100 sommets) en raison de limitations de performance.
    - L'interface graphique est basique et pourrait être améliorée avec des fonctionnalités supplémentaires (zoom, déplacement du graphe, etc.).
  - **Améliorations Possibles** :
    - Ajouter une option pour importer un graphe depuis un fichier.
    - Permettre à l'utilisateur de modifier le graphe interactivement.
    - Optimiser l'algorithme pour gérer des graphes plus grands.
- 

**Auteur** : [Votre Nom]

**Date** : [Date de soumission]

**Classe** : [Nom de la classe ou du cours]