

Rapport 1er Livrable PSI

Connexité et Détection de Cycles dans un Graphe

1. Vérification de la Connexité

Un graphe est connexe si on peut atteindre tous les sommets à partir d'un sommet donné. Pour vérifier cela, on utilise BFS (parcours en largeur) ou DFS (parcours en profondeur).

Méthode appliquée :

On choisit un sommet de départ (ex: sommet 1).
On effectue un BFS ou DFS et on compte les sommets visités.
Si tous les sommets sont atteints, alors le graphe est connexe.

2. Détection de Cycles

Un cycle est une suite de sommets formant une boucle. Pour détecter un cycle, on utilise DFS récursif : Lors du parcours, si on trouve un sommet déjà visité qui n'est pas le parent direct, alors il y a un cycle.

Exemple :
Prenons le graphe avec arêtes



En lançant DFS depuis 1 : on visite 2, puis 3.

Quand on veut aller de 3 à 1, on remarque que 1 est déjà visité et ce n'est pas le parent direct, donc il y a un cycle.

Outils d'IA generative

Notre premier prompt était :

Thought for 20 seconds

Voici un exemple d'outil de visualisation de graphe utilisant **System.Drawing** avec explications des choix techniques :

1. Classe **GrapheVisualiseur** (Code complet)

```
csharp
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;

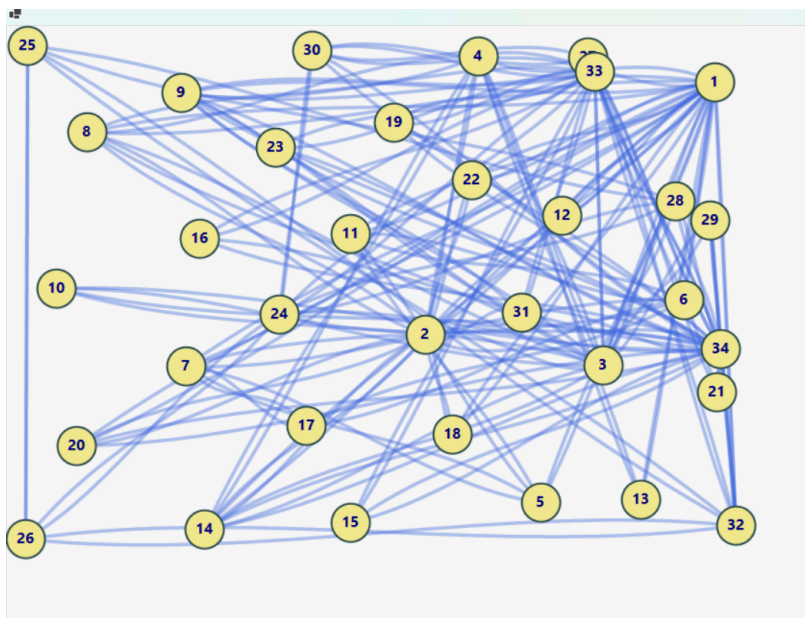
public class GrapheVisualiseur : Form
{
    private Graphe _graphe;
    private PictureBox _pictureBox;
    private Bitmap _bitmap;
    private Dictionary<int, Point> _positions = new Dictionary<int, Point>();
    private const int NODE_RADIUS = 28;

    public GrapheVisualiseur(Graphe graphe)
    {
        _graphe = graphe;
        InitializeComponent();
        GenererPositionsAleatoires();
    }
}
```

Message DeepSeek

DeepThink (R1) Search

AI-generated, for reference only



À l'aide de l'outil DeepSeek nous avons eu cette première version

La première visualisation n'était pas très bonne, certains sommets étaient superposés et ce n'était pas lisible. Suite à des problèmes de serveurs nous avons de changer et utiliser chatGPT Avec ce prompt qui nous donné cette visualisation

je veux aucun chevauchement et que les liens soient plus clairs

Pour éviter les chevauchements et rendre les liens plus clairs, voici quelques améliorations possibles :

1. Utilisation d'un algorithme de placement de nœuds :

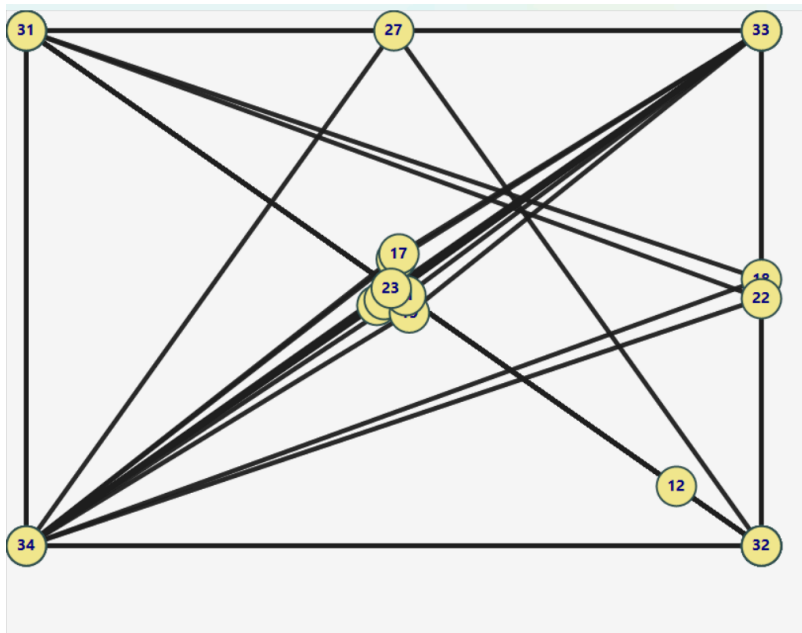
- Remplacer la disposition aléatoire par un algorithme de placement automatique comme *Force-Directed Layout* (Fruchterman-Reingold, Kamada-Kawai).
- Assurer une distance minimale entre les nœuds.

2. Amélioration du dessin des liens :

- Éviter le chevauchement des liens en ajustant leurs courbes avec un placement plus intelligent des points de contrôle.
- Ajouter de la transparence pour mieux visualiser les connexions superposées.
- Utiliser différentes couleurs ou épaisseurs pour différencier les liens proches.

3. Ajustements graphiques :

- Augmenter l'espace autour des nœuds pour réduire le croisement.
- Dessiner les nœuds en premier et les liens ensuite pour éviter que les liens couvrent les labels des nœuds.



Puis de retour sur DeepSeek nous avons fourni le promp suivant

Je veux que le graphe soit comme ça

Le 1 est au centre est autour toutes ces connexions et les connexions autours ont leurs connexions



Thought for 39 seconds

Voici le code complet pour visualiser le graphe avec le nœud 1 au centre et ses connexions organisées en cercles concentriques :

Ce code nous a donc donné cette visualisation du graphe !

