

# Algorithme de Dijkstra

Le plus court chemin



Thanina AIT OUAKLI  
Nadjib BENAMROUCHE

# Plan de travail

**Introduction**

**État d'avancement du projet**

**Problématique et solutions**

**Premier programme**

**Perspectives et planning**

**Conclusion**

# Introduction

En théorie des graphes, l'algorithme du plus court chemin consiste à chercher un accès d'un sommet vers un autre avec un coût moindre.

L'algorithme d'Edsger Dijkstra construit graduellement un meilleur chemin d'un nœud  $u$  vers  $v$  dans un graphe  $G$ , en minimisant à chaque étape les coûts de sauts intermédiaires.

Un autre variant de l'algorithme permet de trouver un chemin à moindres coûts à partir d'un point de départ racine vers n'importe quel nœud de  $G$ .

# État d'avancement

2

- Contexte assimilé : parcours de graphe, problématique, représentation de données, complexité de l'algorithme au meilleur et pire cas.
- Développement d'une première solution: représentation matricielle et par liste du graphe, récursivité.
- Préparer des codes référentiels pour comparer les résultats obtenus.

# Problématique

| La représentation dudit graphe à parcourir est étroitement liée à « l'efficacité » du parcours et donc de la complexité asymptotique.

La problématique est de trouver un meilleur compromis entre complexité, espace mémoire et efficacité algorithmique.

Matrice :  $O(n^2)$ , avec  $n$  noeuds

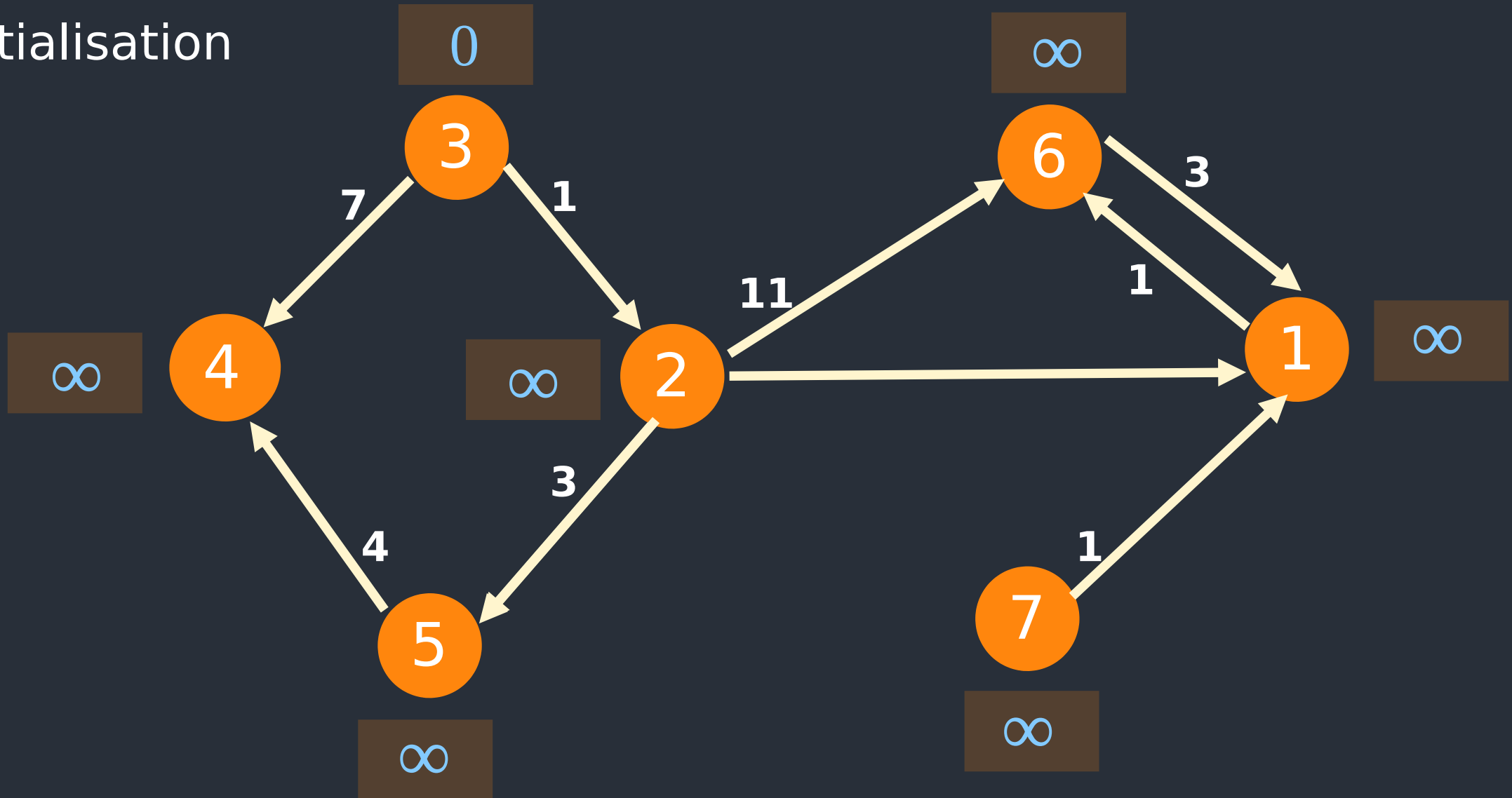
Liste d'adjacence + tas binaire :  $O((m+n) \times \log(n))$ , avec  $m$  arcs

Liste d'adjacence + tas de Fibonacci :  $O(m+n \times \log(n))$

# Premier programme

4

➤ Initialisation



# Perspective et planning

5

**7 dec**

Améliorer le  
code python  
éventuellement  
Tas binaire, Fibo

**9 dec**

Calcul exacte  
de complexité.  
Réaliser des  
tests unitaires

**11 dec**

Mesurer les  
performances.  
En matière de  
temps, espace

**15 dec**

Comparer les  
Résultats, faire  
des vérifications  
sur le travail

**20 dec**

Finaliser le  
projet , rédiger  
le rapport final  
sous Texmaker.

Timeline

# Conclusion

| Le principe de l'algorithme de Dijkstra peut paraître relativement simple au départ, notamment pour les graphes simples (pas de temps, poids positifs) néanmoins, programmer naïvement ce genre d'algorithme peut aboutir rapidement à une performance (complexité) en déclin.

L'avancement du travail est globalement assez bon compte tenu des détails à 'pofiner' et d'autres facteurs relatifs à la gestion d'autres projets.



It's NOT just about ideas  
It's about  
**MAKING IDEAS HAPPEN**

**Merci !**

Dirigé par: Professeur **P**hilippe **L**anglois