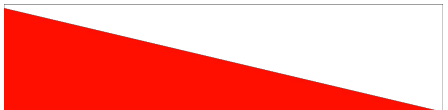
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Soutenance BE Graphes

Nina BATKO - Alexandre DAURIAC

3 MIC D



Université de Toulouse

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Plan de la présentation

I. Introduction et contexte de développement

II. Tests de validité

III. Tests de performance

IV. Problème ouvert

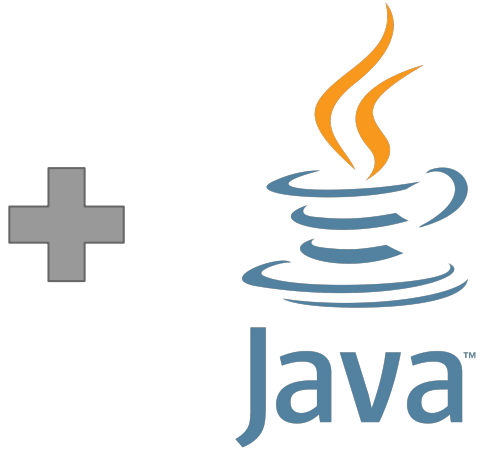
V. Conclusion

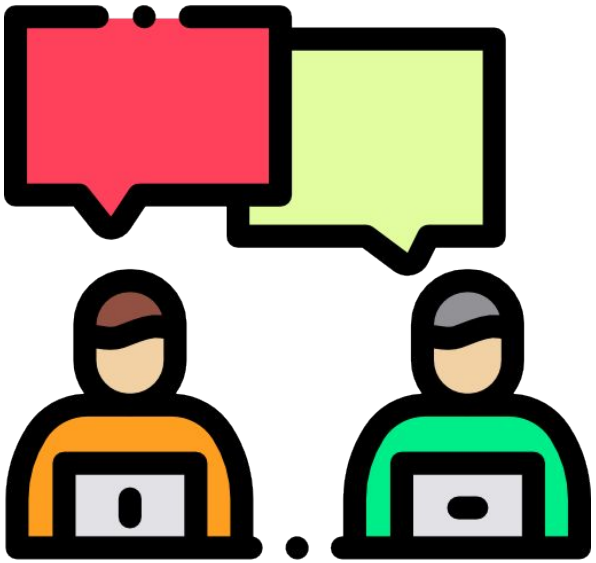
Université de Toulouse

I. Introduction et contexte de développement

- Utiliser la théorie des graphes pour résoudre des problèmes de plus court chemin sur une carte
- Implémenter des algorithmes de Dijkstra et A*
- Implémenter des tests unitaires et des tests de performance
- Comparer l'efficacité de nos algorithmes
- Prendre en compte différents modes de déplacement

I. Introduction et contexte de développement





- Code collaboratif en binôme
- Utilisation de GitHub et Eclipse pour coder un programme en Java
- Utilisation de packages déjà prêts
- Exploitation de JavaDoc

II. Tests de validité

```
Origin node 162 processed.
Node 162 marked.
LabelId : 162 , Marquage : true , Cout : 0.0
Heap size: 0
Number of successors: 2
Node 163 reached.
Heap size: 1
Node 421 reached.
Heap size: 2
Node 421 marked.
LabelId : 421 , Marquage : true , Cout : 22.43400001525879
Heap size: 1
Number of successors: 2
Node 422 reached.
Heap size: 2
```

○
○
○

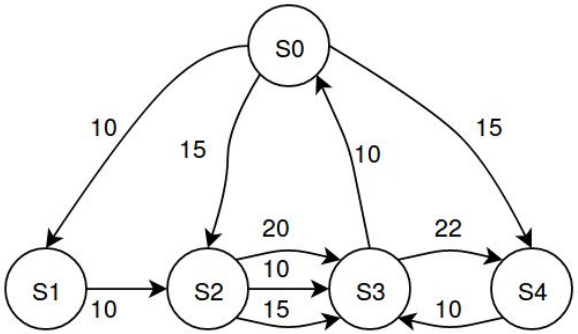
```
Node 442 reached.
Heap size: 9
Node 437 marked.
LabelId : 437 , Marquage : true , Cout : 181.19480919837952
Heap size: 8
Number of successors: 3
Node 444 reached.
Heap size: 9
Destination node 436 reached.
Number of arc: 8, number of iterations: 17
```

Tout d'abord nous avons effectué une vérification visuelle:

- Bon traitement des noeuds parcourus
- Coût croissant (Dijkstra)
- Taille de la pile cohérente
- Bon marquage des noeuds
- Nombre de successeurs cohérent

II. Tests de validité

Test sur un graphe simple




```
---Test fonctionnement Dijkstra en fonction de Bellman Ford---
---------------Affichage : (Sommet pere, Coût)----------------
s0: (--,----) (s0,10.0) (s0,15.0) (s2,25.0) (s0,15.0)
s1: (s3,35.0) (--,----) (s1,10.0) (s2,20.0) (s3,42.0)
s2: (s3,25.0) (s0,35.0) (--,----) (s2,10.0) (s3,32.0)
s3: (s3,15.0) (s0,25.0) (s0,30.0) (--,----) (s3,22.0)
s4: (s4,10.0) (s0,20.0) (s0,25.0) (s2,35.0) (--,----)
```

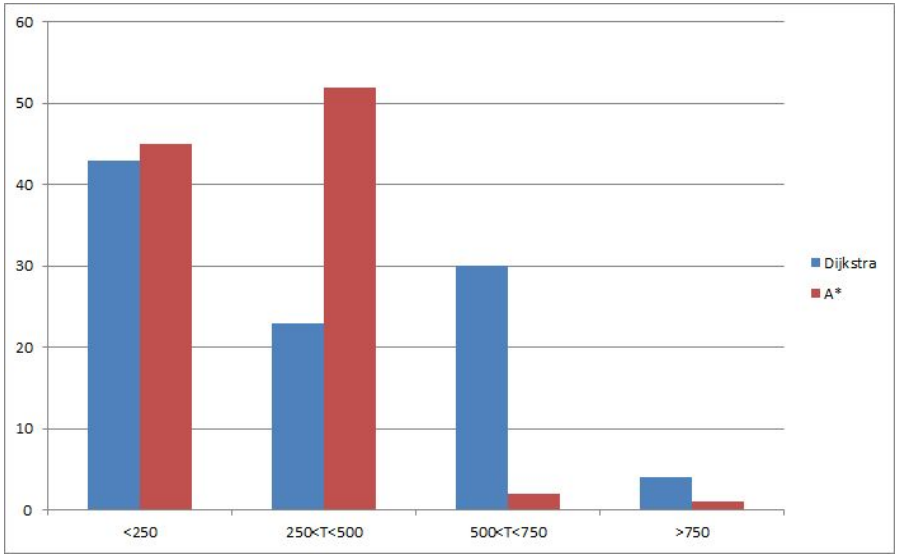
Tests sur une carte (Picardie + Nouvelle-Zélande)

Différents types de chemin :

- Nul
- Court (<10 km)
- Intermédiaire (~100 km)
- Long (~200 km)
- Non connexe

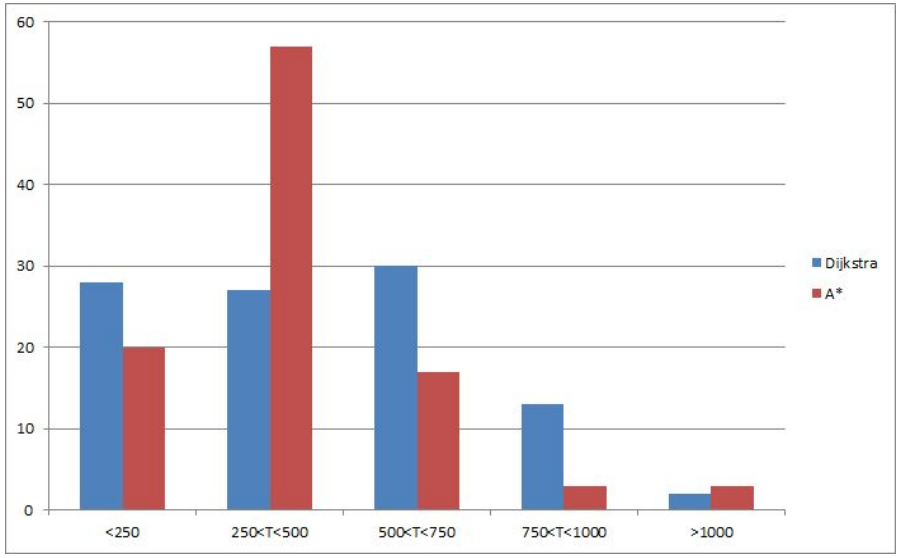
OK en mode Temps
Légers écarts en mode Distance: <1%

III. Tests de performance



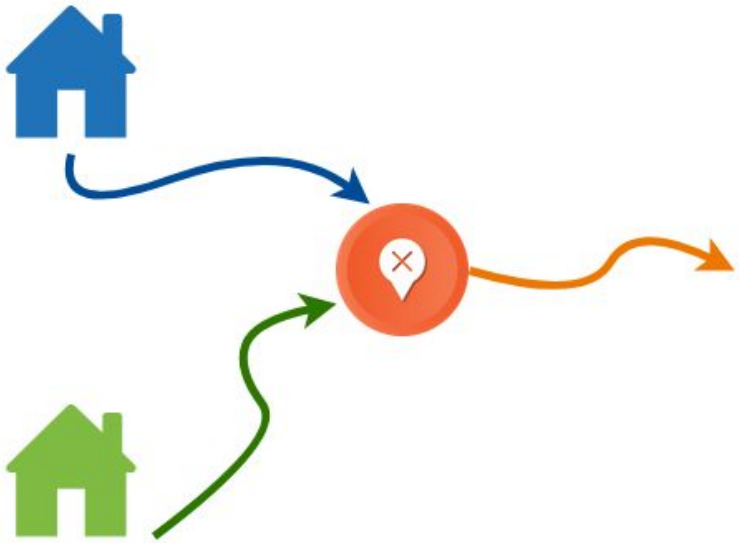
Temps d'exécution (en ms) en mode distance

Dijkstra : 117 850 sommets visités
A* : 23 905 sommets visités



Temps d'exécution (en ms) en mode temps

Dijkstra : 128 044 sommets visités
A* : 58 606 sommets visités



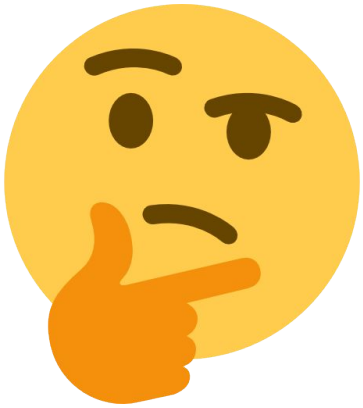
Problème du covoiturage:

Deux personnes partant de leurs origines respectives doivent se rejoindre à une aire de covoiturage pour continuer le chemin ensemble. L'objectif est de minimiser la somme des trajets des deux voitures.

Solution envisagée:

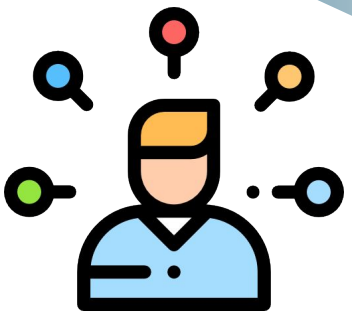
- Nous lançons Dijkstra pour les deux points de départ
- Nous choisissons l'endroit qui minimise le chemin pour les deux voitures
- Nous lançons Dijkstra pour le reste du parcours

V. Conclusion



Problèmes rencontrés:

- Prise en main de git
- Trouver les endroits qui posent problème dans l'algorithme
- Synchroniser le travail en binôme



Nouveaux acquis:

- Apprentissage du code en binôme en parallèle
- Exploitation de packages déjà existants
- Utilisation de nouveaux outils (ex: GitHub)
- Concrétisation de la théorie apprise en cours de Graphes

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Merci pour votre écoute.

Université de Toulouse