Projet IPI: chemins de poids minimum

Romain PEREIRA

04/12/2017

Table des matières

Préambule

Ce projet est réalisé dans le cadre de mes études à l'ENSIIE.
Le but est d'implémenter des algorithmes de recherche de 'chemin le plus court', dans des graphes orientés.
Ce document rapporte mon travail, et explique les choix techniques que j'ai pris.

On considère (X, A) un graphe.

- n : Card(X)
- X : sommets du graphe
- A : arcs du graphe
- s : sommet 'source', celui à partir duquelle les chemins sont construits
- t : sommet 'target', celui vers lequel on souhaite construire un chemin

1 Recherche de chemin le plus court

1.1 Parcours en largeur (graphes non-pondérés)

On considère ici un graphe où les arcs sont non pondérés.
'Le chemin le plus court' entre 2 sommets corresponds au chemin avec un nombre d'arc minimum.
On souhaite savoir s'il y a existence d'un arc ou non entre 2 sommets. Cette information est un booléen, et peut donc être coddé sur un bit. J'économise ainsi beaucoup de mémoire. (8 fois plus que si l'arc était coddé sur un octet), en representant mes arcs comme une suite de bit. De plus, je ne perd pas de temps en lecture / écriture dans le tableau des arcs, et en réduisant la mémoire utilisé, je rends mon programme plus 'cache-friendly' (voir explication détaillé dans le code)

Le p-ième bit vaut 1 ou 0, selon qu'il existe un c

1.2 Algorithme de Dijkstra (graphes pondérés positivement)

On considère ici un graphe où les arcs sont pondérés avec des poids positifs.
'Le chemin le plus court' entre 2 sommets corresponds au chemin avec la somme des poids de ses arcs minimum.

2 Application : resolution labyrinthe