

TP d'algorithmique et structures de données

TP4 – Algorithme de Dijkstra

Exercice 1

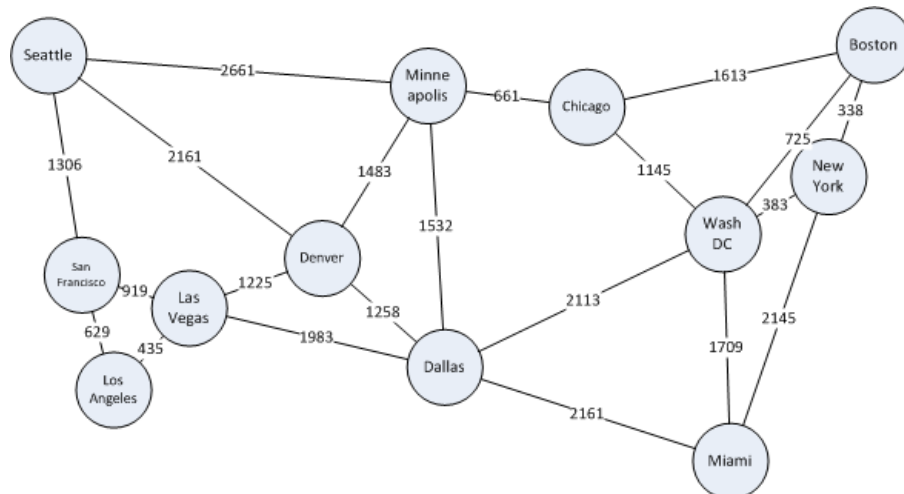
On rappelle ci-dessous l'algorithme de Dijkstra :

Entrée : $\text{Graphe } G = (V, E)$, fonction de coût associé aux arrêtes c et le sommet initial $s_{init} \in V$

1. Initialiser tous les sommets à *non-marqué*. Mettre $L(i) = +\infty$ pour tous les sommets i .
2. Mettre $L(s_{init}) = 0$ et $p(s_{init}) = 0$.
3. Choisir le sommet i non-marqué qui a le label (valeur de $L(i)$) le plus petit.
4. Marquer i .
5. Pour chaque sommet j non-marqué voisin de i si $L(j) > L(i) + c_{ij}$, alors mettre $L(j) = L(i) + c_{ij}$ et $p(j) = i$.
6. Répéter 3-5 jusqu'à ce qu'il n'y a plus de sommet non-marqué.

À faire :

1. Implantez cet algorithme en Javascoll.
2. Testez cet algorithme sur le graphe suivant et afficher les chemins les plus courts entre « Seattle » et toutes les autres villes :



Exercice 2

Étant donné un graphe $G = (V, E)$ qui représente la relation amis, écrivez un programme qui retrouvera le cercle d'amis pour une personne (c.-à-d. la clôture transitive de la relation amis).