TD d'algorithmique avancée

TD 11 : Plus courts chemins pour tout couple de sommets

Jean-Michel Dischler et Frédéric Vivien

Nous nous intéressons ici à la recherche des plus courts chemins entre tous les couples de sommets d'un graphe (typiquement on cherche à élaborer la table des distances entre tous les couples de villes d'un atlas routier). On dispose en entrée d'un graphe orienté $G = (S, A)$ et d'une fonction de pondération w. Nous supposons que le graphe G peut contenir des arcs de poids négatifs mais pas des circuits de poids strictement négatifs. On note n le nombre de sommets de G ($n = |S|$), et on note $\{1, 2, ..., n\}$ les n sommets de G.

Programmation dynamique naïve

Comme nous l'avons remarqué en cours, tout sous-chemin d'un plus court chemin est lui-même un plus court chemin et le problème des plus courts chemins vérifie bien la propriété de sous-structure optimale : nous pouvons donc essayer de le résoudre par programmation dynamique.

1. On note $d_{i,j}^{(m)}$ le poids minimal d'un chemin d'*au plus* m arcs du sommet i au sommet j. Définissez récursivement $d_{i,j}^{(m)}$ (la récursion portera ici sur le nombre d'arcs d'un plus court chemin).
2. Construisez, au moyen de la formule de récurrence obtenue à la question précédente, un algorithme de calcul des longueurs des plus courts chemins pour tout couple de sommets, algorithme basé sur le paradigme de la programmation dynamique.
3. Quelle est la complexité de votre algorithme ?
4. Exécutez votre algorithme sur le graphe de la figure 1.

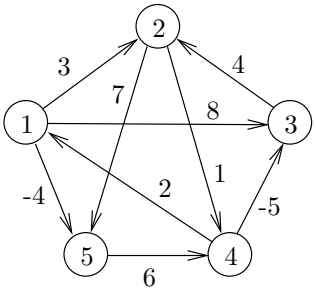

FIG. 1 – Exemple de graphe orienté.

Algorithme de Floyd-Warshall

L'algorithme de Floyd-Warshall est un autre algorithme conçu suivant le principe de la programmation dynamique.

1. Ici la récursion n'a pas lieu sur le nombre d'arcs d'un plus court chemin, mais sur les sommets intermédiaires de ces chemins, un sommet intermédiaire étant un sommet autre que les extrémités du chemin. Ici, $d_{i,j}^{(k)}$ est la longueur du plus court chemin de i à j n'utilisant comme sommets intermédiaires que des sommets parmi $\{1, 2, ..., k\}$.

   Définissez $d_{i,j}^{(k)}$ de manière récursive.

2. Proposez un algorithme de calcul de la longueur des plus courts chemins basé sur la récursion précédente et le paradigme de la programmation dynamique.
3. Quelle est la complexité de votre algorithme ?
4. Proposez un algorithme de construction effective des plus courts chemins à partir de l'algorithme précédent.
5. Exécutez votre algorithme sur le graphe de la figure 1.