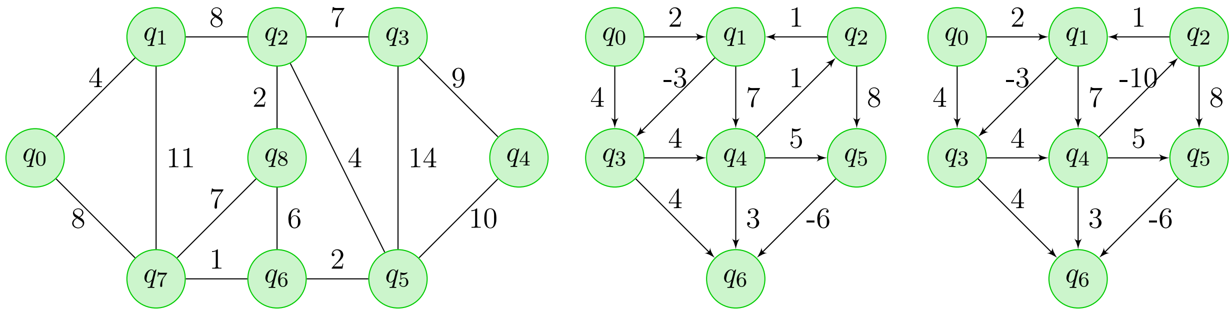
Algorithmique (AL5)

TD n° 5 : plus courts chemins, algorithme de Dijkstra, algorithme de Bellman-Ford

Exercice 1 : Plus courts chemins

Pour chacun des graphes ci-dessous, donner la longueur des plus courts chemins du sommet q_0 à tous les autres sommets du graphe. Vous pouvez simplement faire ce calcul à la main, sans utiliser d'algorithme particulier. Dessiner l'arbre des plus courts chemins issus de q_0.



Exercice 2 : Algorithme de Dijkstra

Appliquer l'algorithme de Dijkstra au premier graphe ci-dessus pour calculer les plus courts chemins. Détailler l'exécution. Que se passe-t-il si on exécute l'algorithme de Dijkstra sur le deuxième graphe ?

Exercice 3 : Algorithme de Bellman-Ford

Appliquer l'algorithme de Bellman-Ford au deuxième graphe ci-dessus pour calculer les plus courts chemins d'origine q_0. Que se passe-t-il si on exécute l'algorithme de Bellman-Ford sur le premier graphe ?

Exercice 4 : Variante de Dijkstra

Supposons que le graphe non-orienté donné en entrée soit pondéré non pas sur les arêtes mais sur les sommets. La longueur d'un chemin est la somme des poids des sommets traversés. Concevoir un algorithme pour résoudre le problème des plus courts chemins isssus d'un sommet s quand tous les poids sont supérieurs ou égaux à zéro. Justifier.

Exercice 5 : Itérations de Bellman-Ford

1. Montrer que Bellman-Ford peut dans certains cas de graphes obtenir un résultat correct dès le premier tour de la boucle principale.
2. Alors, pourquoi en faire d'autres ? Proposer un exemple pire des cas.
3. Modifier l'algorithme de Bellman-Ford pour qu'il s'arrête avant $n - 1$ itérations, si l'on peut.

Utiliser des graphes pour résoudre des problèmes

Exercice 6 :

On souhaite convertir de l'argent d'une devise dans une autre. Le problème est que toutes les conversions ne sont pas possibles : pour deux monnaies A et B, on peut parfois convertir de l'argent de A en B, parfois non. On considère donc un *graphe de change* $G = (V, E)$ entre monnaies donnant les conversions possibles. Ce graphe est orienté (parfois on peut convertir A en B mais pas B en A).

La *fonction de change* est une fonction c telle que

- une somme S en monnaie A vaut $S \cdot c(A, B)$ en monnaie B (taxes éventuelles incluses).
- $c(A, B)$ est défini si et seulement si (A, B) est un arc du graphe de change.
- si $c(A, B)$ est défini, $c(A, B) > 0$

Le *graphe de change étendu* est le graphe $G' = (V, E, c)$ pondéré par la fonction de change. Une *séquence de change* S est la conversion d'une monnaie A_1 en monnaie A_k en passant par les monnaies intermédiaires $A_2 ... A_{k-1}$ (en supposant bien sûr toutes ces conversions possibles). Il lui correspond un chemin dans le graphe de change.

1. Quel est le taux de change de A_1 en A_k dans une séquence de change $A_1, A_2, \dots, A_k$?

2. Dans quelle condition (exprimée sur G') quelqu'un peut-il devenir *infiniment riche* en changeant de l'argent ?

Étant donné deux séquences de change différentes de la monnaie A en la monnaie B, la *meilleure* des deux est celle qui a le taux le plus élevé.

3. Supposons que l'on connaisse une séquence de change S_1 de la monnaie A en la monnaie B, d'une part, et une séquence S_2 de la monnaie A en la monnaie C d'autre part. Supposons que l'arc (C, B) existe, dans le graphe de change. Écrire une condition de *relâchement* en comparant les taux des séquences S_1 d'une part, S_2 puis (C, B) d'autre part, et gardant la meilleure.

4. Écrire une version modifiée de l'algorithme de Bellman-Ford, utilisant cette condition de relâchement modifiée, donnant les meilleurs taux de change d'une monnaie A vers toutes les autres.

Pourquoi est-ce correct ?