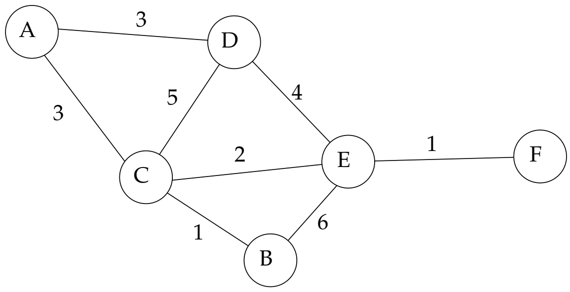
Routage

1 Échauffement – Algorithme de routage à vecteur de distance

On considère le réseau ci-contre.



Question 1.1. *Au temps 0 les tables de distance et de successeur sont stables. La liaison E-F tombe en panne. Faites dérouler l'algorithme pendant 5 itérations.*

Question 1.2. *Même exercice, mais pour la variante de l'algorithme avec clivage d'horizon.*

2 Diffusion généralisée

On souhaite trouver un algorithme de routage pour la diffusion d'un message à tous les nœuds du réseau.

Question 2.1. *Proposez une méthode simpliste, et donner ses défauts.*

Question 2.2. *Montrer que diffuser sans copie inutile revient à construire un arbre de diffusion. Rappelez un algorithme calculant un arbre de poids minimal.*

Question 2.3. *Dans un graphe non-orienté, on considère l'arbre des plus courts chemins (superposition des plus courts chemins de la racine à tous les nœuds), donner une méthode distribuée pour trouver cet arbre, en se basant sur la table de routage des plus courts chemins. Illustrer votre algorithme sur un petit exemple.*

3 Semi-anneaux fermés

Un **semi-anneau fermé** est un structure algébrique $(C, \oplus, \odot, \overline{0}, \overline{1})$, où C est un ensemble d'éléments muni d'un opérateur de **sommation** $\oplus$ et d'un opérateur d'**extension** $\odot$, et où $\overline{0}$ et $\overline{1}$ sont des éléments de C, qui vérifie les propriétés suivantes :

1. $(C, \oplus, \overline{0})$ est un **monoïde** :
   - C est **fermé** pour $\oplus$: pour tout $a, b \in C$, $a \oplus b \in C$
   - $\oplus$ est **associatif**
   - $\overline{0}$ est l'**élément neutre** de $\oplus$: $a \oplus \overline{0} = \overline{0} \oplus a = a$.
2. $(C, \odot, \overline{1})$ est un **monoïde**.
3. $\oplus$ est **commutatif**
4. $\oplus$ est **idempotent** : $a \oplus a = a$ pour tout $a \in C$
5. $\odot$ est **distributif** par rapport à $\oplus$ ($\forall a, b, c \in C, a \odot (b \oplus c) = a \odot b \oplus a \odot c$ et $(b \oplus c) \odot a = b \odot a \oplus c \odot a$),
6. $\overline{0}$ est **absorbant** pour $\odot$ ($\forall a \in C, a \odot \overline{0} = \overline{0} \odot a = \overline{0}$).
7. Si $a_1, a_2, a_3, \ldots$ est une séquence dénombrable d'éléments de C, alors $a_1 \oplus a_2 \oplus a_3 \oplus \cdots$ est bien défini est appartient à C.
8. Associativité, commutativité et idempotence s'appliquent aux sommes infinies, de même que la distributivité de $\odot$ sur $\oplus$.

3.1 Cas des plus courts chemins

On considère un graphe orienté $G = (S, A)$ muni d'une fonction d'étiquetage des arcs $\lambda : S \times S \to C$. On utilise l'associativité de l'opérateur $\odot$ pour étendre aux chemins la notion d'étiquette. L'étiquette du chemin $p = (v_1, v_2, v_3, \ldots, v_k)$ est donc

$$\lambda(p) = \lambda(v_1, v_2) \odot \lambda(v_2, v_3) \odot \cdots \odot \lambda(v_{k-1}, v_k) \quad \text{et} \quad \lambda(\epsilon) = \overline{1} \text{ pour le chemin vide } \epsilon$$

Question 3.1. *Si on considère que l'étiquette d'un arc est son poids, quel doit être l'opérateur $\odot$ pour pouvoir définir le poids d'un chemin ? Quel est son élément neutre ?*

Question 3.2. *Quel doit être l'opérateur de sommation $\oplus$ si on veut que le poids du plus court chemin de i à j soit*

$$l_{i,j} = \bigoplus_{p=i \rightsquigarrow j} \lambda(p)$$

Question 3.3. *Dans le cas général, à quoi correspondent la commutativité et l'associativité de $\oplus$? Quel est l'élément absorbant $\overline{0}$ et son rôle dans ce cas ?*

Question 3.4. *À quoi correspondent l'idempotence de $\oplus$ et la distributivité de $\odot$ sur $\oplus$? Pourquoi est-on amené à considérer des séquences infinies dénombrables ?*

3.2 Algorithme générique

On propose ici une généralisation de l'algorithme de Floyd-Warshall pour calculer les $l_{i,j}$ par programmation dynamique. Soit $Q_{i,j}^{(k)}$ l'ensemble des chemins de i à j dont tous les sommets intermédiaires se trouvent dans l'ensemble $\{1, 2, \ldots, k\}$. On définit

$$l_{i,j}^{(k)} = \bigoplus_{p \in Q_{i,j}^{(k)}} \lambda(p)$$

Question 3.5. *Donner une formule de récurrence pour calculer $l_{i,j}^{(k)}$ en fonction des $l_{i,j}^{(k-1)}$. Donner les $l_{i,j}^{(0)}$.*

Question 3.6. *Généraliser l'approche de programmation dynamique de Floyd-Warshall pour calculer les $l_{i,j}^{(k)}$.*

3.3 Adaptation aux arbres de poids minimaux.

On considère un graphe connexe non-orienté $G = (S, A)$ muni d'une fonction de pondération des arêtes $w : A \to \mathbb{R}$. On suppose que tous les poids des arêtes $w(i, j)$ sont uniques. On cherche un arbre couvrant de poids minimal. Dans ce cas, on peut montrer que l'arbre couvrant est unique.

On définit pour chaque couple de sommets $i, j \in S$ la pondération **minimax** :

$$m_{i,j} = \min_{p=i \rightsquigarrow j} \left\{ \max_{\text{arête } e \text{ du chemin } p} \{w(e)\} \right\}$$

Question 3.7. *Montrer que $(\mathcal{R} \cup \{-\infty, \infty\}, \min, \max, \infty, -\infty)$ est un semi-anneau fermé.*

Question 3.8. *On veut se servir de la procédure de la section 3.2 pour calculer les $m_{i,j}$. Donner la formule de récurrence des $m_{i,j}^{(k)}$.*

Question 3.9. *Soit $T_m = \{(i, j) \in A \text{ tel que } w(i, j) = m_{i,j}\}$. Démontrer que les arcs de T_m forment un arbre couvrant de G.*

Question 3.10. *Montrer que T_m est l'arbre couvrant de poids minimal.*