

## 1 Problématique

Les protocoles de routage (RIP, OSPF) calculent les plus courts chemins entre les routeurs. Les algorithmes utilisés ont été découverts à la fin des années 60 et sont utilisés dans de nombreuses situations.

Comment fonctionnent les algorithmes de plus court chemin ?

## 2 Retour des graphes

### 2.1 Graphe orienté

Un graphe **orienté** possède les mêmes caractéristiques que celles déjà évoquées pour un graphe non orienté, mais dont les arcs ne peuvent être parcourus que dans un sens. On peut ainsi définir un graphe par *une matrice d'adjacence* ou *un dictionnaire d'adjacence*.

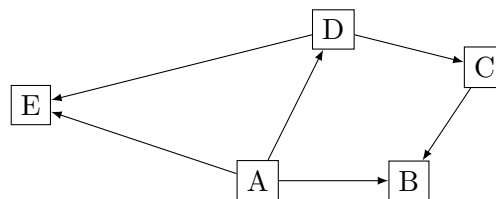


FIGURE 1 – graphe orienté

### À retenir

Pour chaque nœud on peut définir ses **prédécesseurs** et ses **successeurs**.

- Le nœud A ne possède pas de *prédécesseur*.
- Le nœud E ne possède pas de *successeur*.

### Activité 1 :

1. Établir la matrice d'adjacence du graphe figure 1.
2. Établir le dictionnaire d'adjacence du graphe figure 1.
3. Déterminer un cycle dans le graphe.

### 2.2 Graphe pondéré

Les arcs d'un graphe **pondéré** possèdent une *valeur ou poids*. Cette valeur peut être négative.

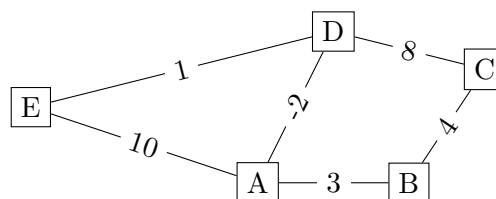


FIGURE 2 – graphe non orienté pondéré

### Activité 2 :

1. Établir la matrice d'adjacence du graphe figure 2.
2. Établir le dictionnaire d'adjacence du graphe figure 2.

### 3 Protocole RIP : algorithme de Bellman-Ford

#### 3.1 Principe

Publié entre 1956 et 1958, cet algorithme calcule les plus courts chemins entre un *nœud source* et tous les autres nœuds d'un graphe orienté et pondéré. Le principe est le suivant :

*La distance pour atteindre chaque nœud correspond à la distance pour atteindre son prédécesseur à laquelle on ajoute le poids de l'arête les séparant.*

La démarche est récursive : on applique le même principe au prédécesseur. Enfin un nœud pouvant avoir plusieurs prédécesseurs, on gardera le chemin le plus court.

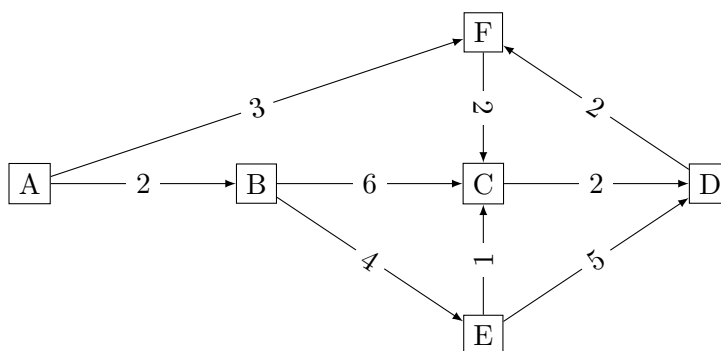


FIGURE 3 – graphe non orienté pondéré

#### Remarque

Dans le graphe figure 3 les pondérations représentent un nombre de routeurs traversés pour atteindre le nœud (routeur) suivant.

#### 3.2 Mise en application

Chaque routeur exécute l'algorithme pour calculer les plus courts chemins vers chaque autre routeur du réseau. L'algorithme détaillé pour le routeur A est le suivant :

```

1 Créer un tableau des distances entre A et les routeurs (A inclus),
  initialisées à l'infini.
2 Dans le tableau modifier la distance vers A à 0.
3
4 Pour chaque sommet
5   Pour chaque arc du graphe
6     Si (la distance du routeur) > (la distance de son prédécesseur +
      poids entre les deux routeurs)
7       La distance du routeur est remplacée par cette nouvelle valeur

```

Code 1 – Algorithme de Bellman Ford

Pour le graphe figure 3, le tableau est initialisé comme suit :

A	B	C	D	E	F
0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

On effectue une première itération sur les arcs.

La distance calculée pour atteindre B correspond à la distance (du tableau) de son prédécesseur (A) augmentée du poids de l'arc A-B :

$$0 + 2 = 2 < \infty$$

Le tableau devient :

A	B	C	D	E	F
0	2	$\infty$	$\infty$	$\infty$	$\infty$

**Activité 3 :** Continuer de dérouler l'algorithme sur le graphe figure 3.

### 3.3 Complexité

La complexité de l'algorithme (code 1) dépend :

- du nombre de sommets (notée S) : on visite chaque sommet (ligne 4) ;
- du nombre d'arcs (notée A) : pour chaque sommet on regarde tous les arcs du graphe (ligne 5).

$$O(S.A)$$

## 4 OSPF : algorithme de Dijkstra

### 4.1 Principe

Il a été publié en 1959 par le néerlandais Edsger Dijkstra.

### 4.2 Mise en application

### 4.3 Complexité