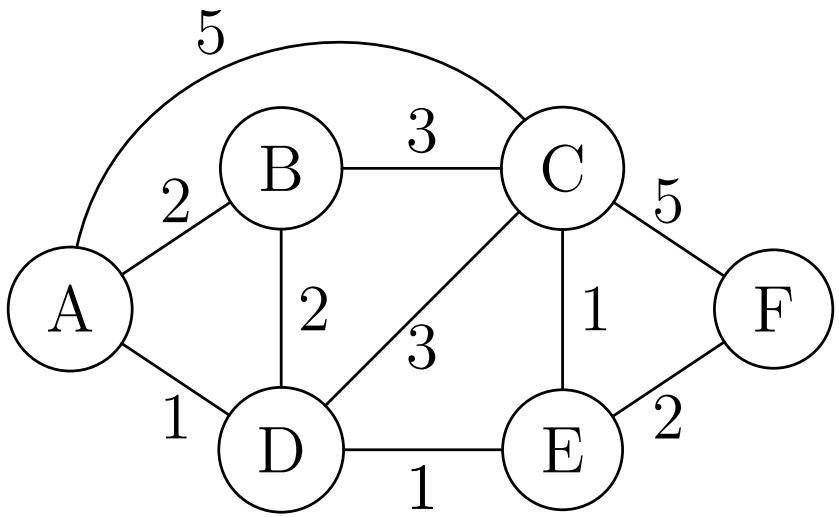
Algorithme de Bellman-Ford

Entrée : $\begin{cases} \text{Un graphe } G = (V, E) \text{ avec une source } s \\ \text{Une fonction de poids } w : E \to \mathbb{R}^+ \end{cases}$

Sortie: $\begin{cases} \text{Un vecteur distance } d \\ \text{Une fonction père } \pi : V \to V \end{cases}$

1. Initialisation des variables d et π
   - 1.1 $d[s] \leftarrow 0$; $\pi[s] \leftarrow s$
   - 1.2 Pour chaque sommet v de V faire $\begin{cases} d[v] \leftarrow \infty \\ \pi(v) \leftarrow NIL \end{cases}$
2. Calcul des distances et de l'arbre de plus court chemin
   - 2.1 Pour $i \leftarrow 1$ à $|V| - 1$ faire
     Pour chaque arête $(u, v) \in E$ faire
     Si $(d[v] > d[u] + w(u, v))$ alors $\begin{cases} d[v] \leftarrow d[u] + w(u, v) \\ \pi(v) \leftarrow u \end{cases}$
3. retourner d et π

Exemple



	A	B	C	D	E	F
init	$(0, A)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$
1	$(0, A)$	$(2, A)$	$(5, A)$	$(1, A)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$
2	$(0, A)$	$(2, A)$	$(4, D)$	$(1, A)$	$(2, D)$	$(10, C)$
3	$(0, A)$	$(2, A)$	$(3, E)$	$(1, A)$	$(2, D)$	$(4, E)$

les couples correspondent à $(d(.), \pi(.))$

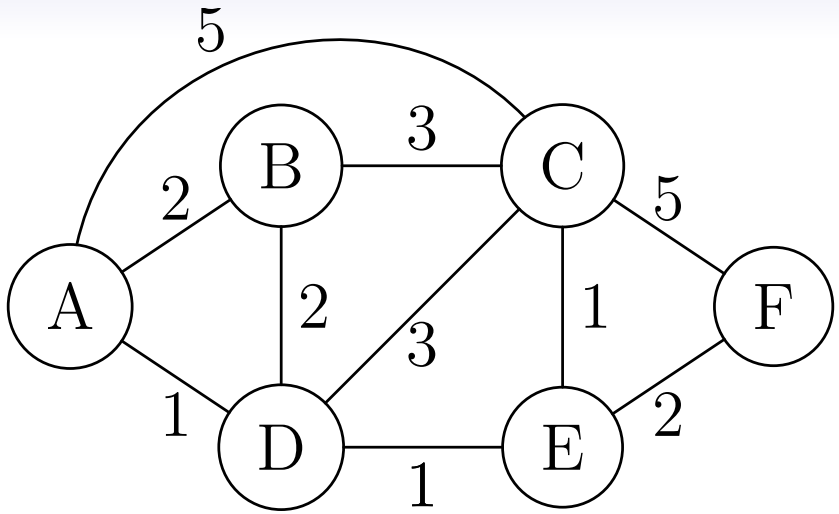
Algorithme de Dijkstra

Entrée : $\begin{cases} \text{Un graphe } G = (V, E) \text{ avec une source } s \\ \text{Une fonction de poids } w : E \to \mathbb{R}^+ \end{cases}$

Sortie: $\begin{cases} \text{Un vecteur distance } d \\ \text{Une fonction père } \pi : V \to V \end{cases}$

1. Initialisation de la source s
   - 1.1 $d[s] \leftarrow 0$; $\pi[s] \leftarrow s$
   - 1.2 Pour chaque sommet v de V faire $\begin{cases} \pi(v) \leftarrow NIL \\ d(v) \leftarrow \infty^+ \end{cases}$
2. $\mathcal{Q} \leftarrow V$; $\mathcal{S} \leftarrow \emptyset$;
3. Tant que $(\mathcal{Q} \neq \emptyset)$ faire
   - 3.1 $u \leftarrow$ Extraire-Le-Minimum$(\mathcal{Q}, d)$;
   - 3.2 $\mathcal{S} \leftarrow \mathcal{S} \cup \{u\}$;
   - 3.3 Pour chaque sommet v voisin de u faire
     Si $(d[v] > d[u] + w(u, v))$ alors $\begin{cases} d[v] \leftarrow d[u] + w(u, v) \\ \pi(v) \leftarrow u \end{cases}$
4. retourner d et π

Exemple



les couples correspondent à $(d(.), \pi(.))$

	$\mathcal{Q}$	A	B	C	D	E	F
I	$\{ABCDEF\}$	$(0, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$
1	$\{BCDEF\}$	$(0, A)$	$(2, A)$	$(5, A)$	$(1, A)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$
2	$\{BCEF\}$	$(0, A)$	$(2, A)$	$(4, D)$	$(1, A)$	$(2, D)$	$(\infty, \emptyset)$
3	$\{CEF\}$	$(0, A)$	$(2, A)$	$(4, D)$	$(1, A)$	$(2, D)$	$(\infty, \emptyset)$
4	$\{CF\}$	$(0, A)$	$(2, A)$	$(3, E)$	$(1, A)$	$(2, D)$	$(4, E)$
5	$\{F\}$	$(0, A)$	$(2, A)$	$(3, E)$	$(1, A)$	$(2, D)$	$(4, E)$
6	$\emptyset$	$(0, A)$	$(2, A)$	$(3, E)$	$(1, A)$	$(2, D)$	$(4, E)$

Complexité de l'algorithme: $O(|V|^2)$

Plus court chemin dans un graphe

- Entrée : Un graphe orienté $G = (N, A)$ où chaque arc possède une longueur non-négative. Un des noeuds du graphe est appelé source.
- Problème: Trouver la longueur du plus court chemin entre toutes les paires de noeuds.
- On suppose que $N = \{1, \ldots n\}$.
- On suppose aussi que G est donné sous forme de matrice $L[1..n, 1..n]$ (on peut prendre $L[i, j] = \infty$ lorsqu'il n'y a pas d'arête entre i et j)
- Algorithme de Floyd: cet algorithme construite une matrice D qui donne la longueur du plus court chemin entre chaque paire de noeuds.

Principe

1. On initialise D à L
2. Après l'itération k, D donne la longueur du plus court chemin lorsque l'on utilise que des noeuds dans $\{1, .., k\}$ comme noeuds intermédiaires (ou éventuellement aucun noeud intermédiaire).

Définition: D_k est la matrice D après l'itération k.
Le résulat final recherché est D_n.

Récurrence

$$D_k[i,j] = \min(D_{k-1}[i,j], D_{k-1}[i,k] + D_{k-1}[k,j])$$

Exercice: expliquer pourquoi.

Floyd(L[1..n,1..n])

Procédure Floyd(L[1..n,1..n])

- D:=L
- Pour $k := 1$ à n
  - Pour $i := 1$ à n
    - Pour $j := 1$ à n
    - $D[i,j] = min(D[i,j], D[i,k] + D[k,j])$
- Retourner D.

Temps dans $O(n^3)$.