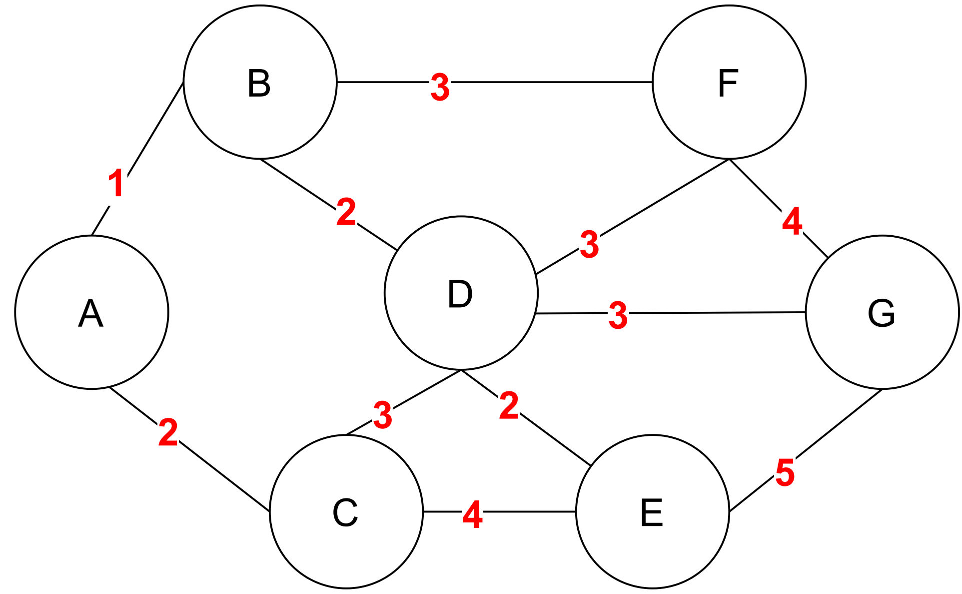
Algorithme du plus court chemin : Dijkstra

Les points représentent les villes il y a donc 7 étapes

on souhaite trouver le chemin le plus court entre 2 villes A et G

on utilise l'algorithme de Dijkstra

le plus court chemin est donc de 6 km et il passe par les points A B D G



A	B	C	D	E	F	G	ETAPES
0	1A	2A					**1**
X	**1A**		3B		4 B		**2**
X	X	**2A**	5C	6C			**3**
X	X	X	**3B**	5D	6D	6D	**3**
X	X	X	X		**4B**	8F	**5**
X	X	X	X	**5D**	X	10 E	**6**
X	X	X	X	X	X	**6D**	**7**

Résultats
1A
3B
6D
G

Définition 3.7 (algorithme de Dijkstra-Moore) *Soit un graphe orienté valué $G = (S, A, f)$, d'ordre n et de taille m, et x un sommet de G. L'algorithme de Dijkstra calcule deux matrices de taille $1 \times n$*

- *$Dist$ matrice des distances telle que $Dist(y)$ = distance optimale de x à y*
- *$Pred$ matrice des prédécesseurs telle que $Pred(y)$ = prédécesseur de y dans le chemin optimal depuis x*

Pour le plus court chemin l'algorithme s'écrit :

fonction $[Dist, Pred] = \text{DIJKSTRA}(G, s)$
Initialisation :
n = nombre de sommets de G
$Pred$ = tableau des prédécesseurs initialisé à 0
$Dist$ = tableau des distances initialisé à $+\infty$ (sauf $Dist(s) = 0$)
W = matrice des poids des arcs (∞ si l'arc n'existe pas)
$C = \{1; 2; \ldots; n\}$ (liste des sommets restant à traiter)
$D = \emptyset$ (liste des sommets déjà traités)
Traitement :
tant que $C \neq \emptyset$ **faire**
  x = sommet de C le plus proche de s
  retirer x de C et le mettre dans D
  pour tout sommet $y \in C$ **faire**
    si $Dist(x) + W(x, y) < Dist(y)$
      alors modifier $Dist(y)$ et $Pred(y) = x$
    fin
  fin faire
fin faire