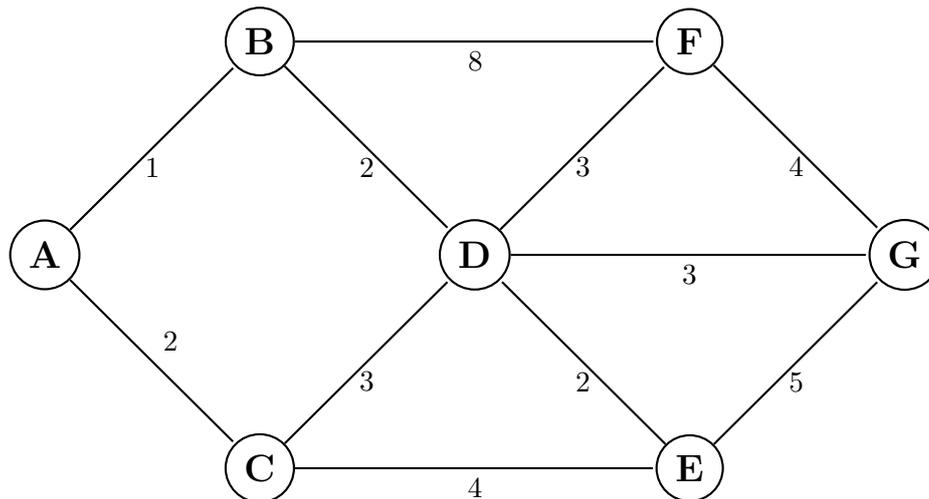


On considère le graphe pondéré non-orienté suivant. On supposera que le poids de chacune des arêtes est positif et on parlera de longueur de chemin par commodité.



**Question :** Quelle est la plus courte distance pour relier  $A$  à  $G$ ? Et quel chemin emprunté pour relier  $A$  à  $G$  de façon optimale?

*Edsger Dijkstra : (1930 - 2002) :* mathématicien et informaticien néerlandais, il développa l'algorithme que nous allons étudier et qui porte son nom.

### Algorithme : trouver la distance

#### Outils

On va créer et compléter

- ▷ une liste contenant les sommets dont on a trouvé la plus courte distance depuis  $A$  ( $\sim$  le départ) : on les qualifiera de "complétés".
- ▷ un dictionnaire des distances contenant :
  - ▷ clé : le sommet
  - ▷ value : la plus courte distance entre  $A$  et le sommet considéré **pour le moment trouvée**

#### Évolution de l'algorithme

A chaque étape :

- ▷ on prend le sommet  $S$  dans le dictionnaire le plus proche de  $A$  qui n'est pas dans la liste de "complétés"
- ▷ on l'ajoute à la liste des "complétés"
- ▷ on étudie tous les sommets voisins  $s_i$  de  $S$  : pour chacun on calcule sa distance au sommet de départ en passant par  $S$  comme :

$$\text{distance entre } A \text{ et } S + \text{distance entre } S \text{ et } s_i$$

2 cas alors :

1. le sommet voisin n'est pas dans le dictionnaire  $\rightarrow$  on l'ajoute avec comme valeur la distance entre  $A$  et le sommet passant par  $S$
2. le sommet voisin est déjà dans le dictionnaire. On compare alors la distance enregistrée dans le dictionnaire et la distance calculée en passant par  $S$ . On garde alors la plus petite des deux.

L'algorithme se termine lorsque le sommet d'arrivée arrive dans la liste des "complétés".

| **Remarque** : On présente souvent l'algorithme de Dijkstra à l'aide d'un tableau, on préférera ici un approche axée à l'implémentation en Python.

| **Application 1** : Appliquer l'algorithme de Dijkstra avec le graphe précédent pour trouver la plus courte distance entre A et G.

## Trouver le plus court chemin

On a trouvé la distance la plus courte entre deux sommets, reste à trouver le chemin qui y mène. Pour cela on reprend l'algorithme précédent et on ajoute un nouvel outil : un dictionnaire des sommets d'accès avec

- ▷ clé : sommet  $\mathcal{S}$
- ▷ valeur : sommet qui permet d'accéder à  $\mathcal{S}$  par le plus court chemin

Dans l'algorithme, lorsqu'on ajoute ou qu'on modifie un élément du dictionnaire des distances, on fait de même dans celui des chemins d'accès en mettant comme valeur le sommet d'accès associé à la nouvelle distance estimée.

**Application 2** : Trouver le plus court chemin, ainsi que la distance, entre les sommets A et G du graphe suivant.

