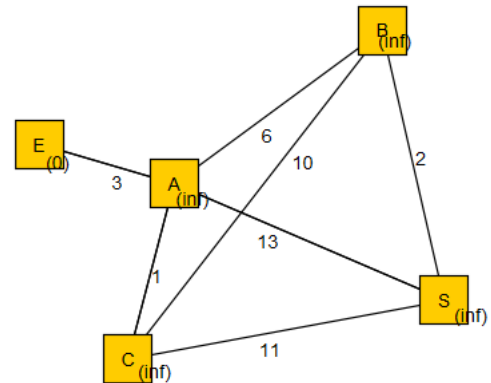


# Exemple d'exécution de l'algorithme de Dijkstra

On ici souhaite trouver le chemin le plus court entre E et S.

## 1<sup>ère</sup> étape : initialisation.

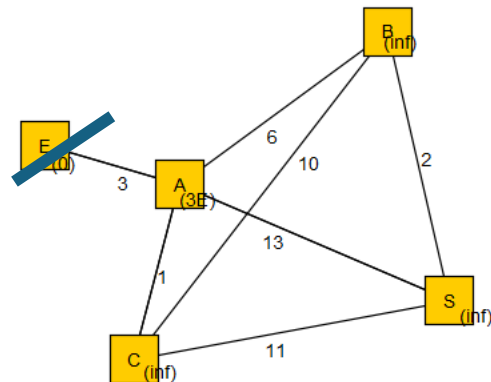
On met tous les **poids à l'infini** (on ne connaît pas encore les chemins) sauf celui du nœud d'entrée E mis à zéro (c'est normal, il n'y a pas de distance entre E et E).



## 2<sup>ème</sup> étape : traitement du sommet de départ E

Le seul chemin disponible est celui de E vers A : on actualise la **valeur** du nœud A en **3E** (la lettre indique le prédécesseur).

Le nœud E a été traité (symbolisé par le trait barrant le nœud), on passe au **nœud A**.

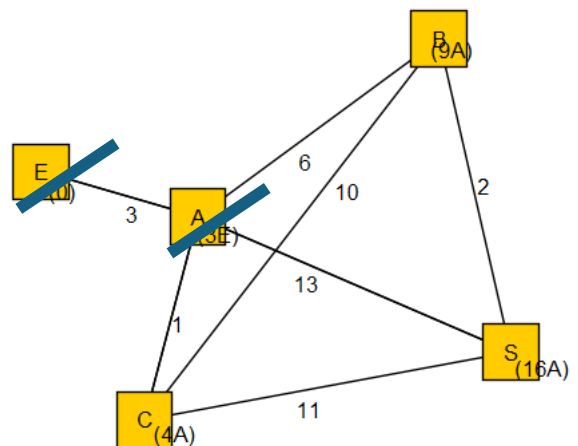


## 3<sup>ème</sup> étape : traitement du nœud A

Il y a trois successeurs possibles :

- de **A vers B**, il y a un poids total de 3 + 6 soit 9. On actualise la valeur du nœud B en **9A**.
- de **A vers C**, valeur du nœud C actualisée en **4A**.
- de **A vers S**, valeur du nœud S actualisée en **16A**.

Le nœud A a été traité, on passe au **nœud C** car il a le poids le plus faible (4A).

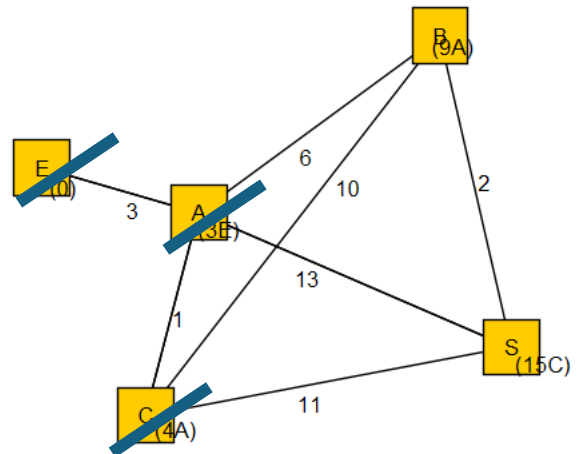


#### 4<sup>ème</sup> étape : traitement du nœud C

Il y a deux successeurs possibles :

- de **C vers B**, il y a un poids total de  $4 + 10$  soit  $14$ .  
On ne **change rien** car  $14 > 9$ .
- de **C vers S**, valeur du nœud C actualisée en **15C**.

Le nœud C a été traité, on passe au **nœud B** car il a le poids le plus faible (9A).



#### 5<sup>ème</sup> étape : traitement du nœud B

Il y a un successeur possible :

- de **B vers S**, il y a un poids total de  $9 + 2$  soit  $11$ .  
On actualise la valeur du nœud S en **11B**.

**Tous les nœuds ont été traités, l'algorithme est terminé.**

En remontant à partir de S, on en déduit le chemin le plus court : S ; B ; A ; E soit E ; A ; B ; S pour un poids total de 11.

