# Théorie des graphes

## Chapitre 4 : Parcours de graphes, algorithme de Dijkstra

5 janvier 2021

INP TOULOUSE ENSEEIHT n7

On considère dans cette partie des graphes pondérés, i.e. à chaque arête/arc est associé un coût positif ou nul. On suppose toujours que le graphe est simple (pas de boucle, pas d'arêtes/arcs multiples).

L'**algorithme de Dijkstra** détermine pour un graphe connexe le plus court chemin d'un sommet choisi à n'importe quel autre sommet du graphe.

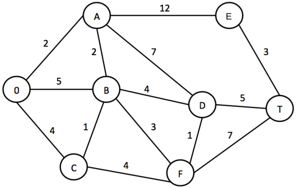
- Cet algorithme peut s'exécuter sur des graphes orientés ou non.
- Il construit un arbre couvrant du graphe, admettant le sommet d'origine comme racine.
- C'est un algorithme en temps polynomial.

- On suppose que le sommet de départ est le sommet 0.
- On prend un vecteur $C$ de taille $n$, $C(i)$ contient la distance courante entre le sommet 0 et le sommet $i$.
- Un ensemble $S$ contient les sommets déjà visités, un autre $R$ contient les sommets restants.

**Initialisation**
**si** l'arête $\{v_0, v_i\}$ existe **alors**
    $C(i) = c_{0i}$
**sinon**
    $C(i) = +\infty$
**fin si**
$S = \{v_0\}$
$R = \{v_1, \ldots, v_n\}$
$i = 0$
**Corps**
**tant que** $R \neq \emptyset$ **faire**
    Choisir $i$ tel que $i = \mathrm{argmin}_{i \in R} C(j)$
    $\{$Mise à jour de $C\}$
    **pour** Tous les sommets $v_j$ voisins de $v_i$ **faire**
        $C(j) = \min(C(j), C(i) + c_{ij})$
    **fin pour**
    Ajouter $v_i$ à $S$
    Retirer $v_i$ à $R$
**fin tant que**

- Argument de terminaison : on visite les sommets un à un, donc $\#R$ est strictement décroissant.
- C'est un algorithme de type glouton qui mène à une solution globale optimale.
- Pour la complexité, on parcourt les sommets (passage d'un sommet de $R$ à $S$ et les arêtes sont toutes parcourues –les arcs 1 fois ; les arêtes 2 fois–). A chaque passage sur un sommet, on chercher le minimum dans $R$. Au pire, la recherche du minimum est linéaire, donc $O(m + n + n^2) = O(m + n^2)$, on peut aussi faire mieux en gardant $R$ trié, mais du coup, il faut pour chaque visite d'arête, insérer dans la liste triée $O((m + n)ln(n))$.

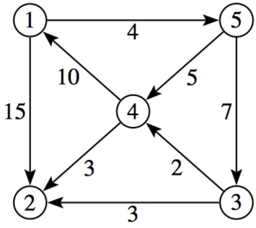
**Exercice 4.1.1**. Trouver le plus court chemin de 0 à $T$




## Solution

| O | A | B | C | D | E | F | T | $S$ | $v_i$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | $\underline{2}_O$ | $5_O$ | $4_O$ | $+\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | $S=\{O\}$ | $v_1=A$ |
| | | $\underline{4}_A$ | $4_O$ | $9_A$ | $14_A$ | $+\infty$ | $+\infty$ | $S=\{O,A\}$ | $v_2=B$ |
| | | | $\underline{4}_O$ | $8_B$ | $14_A$ | $7_B$ | $+\infty$ | $S=\{O,A,B\}$ | $v_3=C$ |
| | | | | $8_B$ | $14_A$ | $\underline{7}_B$ | $+\infty$ | $S=\{O,A,B,C\}$ | $v_4=F$ |
| | | | | $\underline{8}_B$ | $14_A$ | | $14_F$ | $S=\{O,A,B,C,F\}$ | $v_5=D$ |
| | | | | | $14_A$ | | $\underline{13}_D$ | $S=\{O,A,B,C,F,D\}$ | $v_6=T$ |
| | | | | | $\underline{14}_A$ | | | | |

**Exercice 4.1.2**. Trouver les plus courts chemins partant de 1 aux autres sommets du graphe




## Solution

2 : 12 ; 3 :11 ; 4 :9 ; 5 :4.

| 1 | 2 | 3 | 4 | 5 | $S$ | $v_i$ |
|---|---|---|---|---|---|---|
| 0 | $15_0$ | $+\infty$ | $+\infty$ | $\underline{4}_0$ | $S = \{1\}$ | $v_1 = 5$ |
| | $15_0$ | $11_5$ | $\underline{9}_5$ | | $S = \{0, 4\}$ | $v_2 = 9$ |
| | $12_4$ | $\underline{11}_5$ | | | $S = \{0, 4, 9\}$ | $v_3 = 3$ |
| | $\underline{12}_4$ | | | | | |

**Remarque 4.1.1**. si les poids peuvent être négatifs, il faut utiliser l'algorithme de Bellman-Ford, capable d'identifier un cycle absorbant (cycle tel que $d(v_i, v_i) < 0$).