CTD4 : Parcours de graphe

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

Dijkstra L'algorithme de Dijsktra 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 donc 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 tel $C(i)$ 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
$R = \{v_0, v_1, \ldots, v_n\}$
$C(0) = 0$
$C(i) = +\infty \; i \neq 0$
$i = 0$
Corps
tant que $R \neq \emptyset$ **faire**
  Choisir v_i tel que $v_i = \text{argmin}_{v_j \in R} C(j)$
  {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

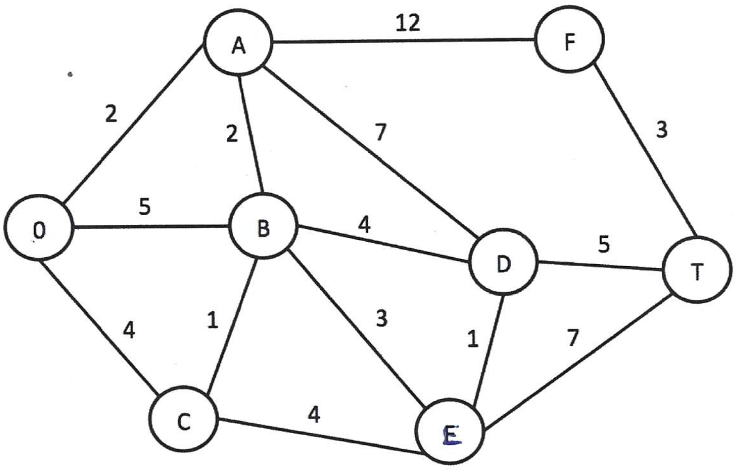
Argument de terminaison : on visite les sommets un à un, donc $\#R$ est strictement décroissant.

C'est un algorithme de type glouton qui mène à une solution globale optimale.

Quelle complexité ?

On parcourt les sommets (passage de 1 sommet R à S et les arêtes sont toutes parcourues - les arcs 1 fois, les arêtes 2 fois -) A chaque passage sur un sommet, on cherche le min dans R. Au pire, la recherche du min est

▷ **Exercice 20** *Trouver le plus court chemin de* 0 *à* T



Initialisation :

$R = \{A, B, C, D, E, F, T\}$.

$$\begin{array}{cccccccccc} & 0 & A & B & C & D & E & F & T & \\ C_0 = [& \cancel{0}, & 2, & 5, & 4, & +\infty, & +\infty, & +\infty, & +\infty &] \end{array}$$

$A = \operatorname{argmin} C_0$

$$\begin{array}{cccccccccc} & 0 & A & B & C & D & E & F & T & \\ C_1 = [& \cancel{0}, & \cancel{2}, & \underline{4}, & 4, & \underline{9}, & +\infty, & \underline{14} & +\infty &]. \end{array}$$

$B = \operatorname{argmin} C_1$ (ou C).

$$\begin{array}{cccccccccc} & 0 & A & B & C & D & E & F & T & \\ C_2 = [& \cancel{0} & \cancel{2} & \cancel{4} & \underline{4} & \underline{8} & \underline{7} & 14 & +\infty &]. \end{array}$$

$C = \operatorname{argmin} C_2$

$C_3 = C_2 \setminus C$

$E = \operatorname{argmin} C_3$.

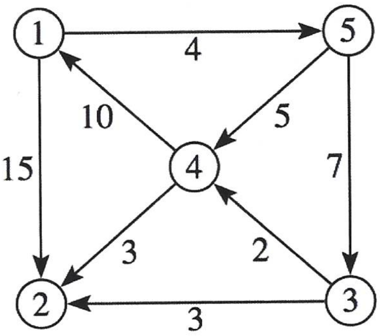
$C_4 = [\cancel{0}\ \cancel{2}\ \cancel{4}\ \cancel{4}\ \underline{8}\ \cancel{7}\ 14\ \underline{14}]$.

$\operatorname{argmin} C_4 = D$

$C_5 = [\cancel{0}\ \cancel{2}\ \cancel{4}\ \cancel{4}\ \cancel{8}\ \cancel{7}\ 14\ \underline{13}]$.

$T = \operatorname{argmin} C_5$

▷ **Exercice 21** *Trouver les plus courts chemins partant de* 1 *aux autres sommets du graphe*



Remarque : si les poids peuvent être négatifs, il faut utiliser l'algorithme de Bellman-Ford, capable d'identifier un cycle absorbant.