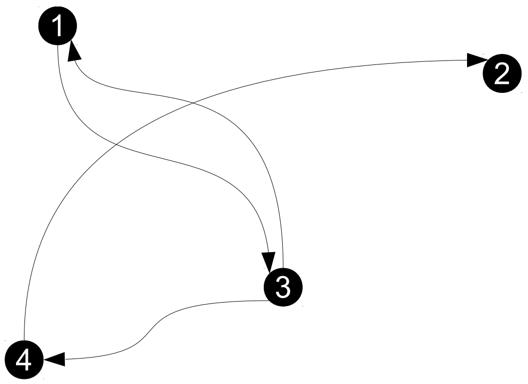
Document de travail pour le TD6
Recherches de plus court chemins dans les graphes (partie 2)

Vocabulaire vu au TD5 : graphe, graphe valué, arc réflexif, chemin, plus court chemin, cycle, cycle négatif (ou absorbant).

Question 1- graphes et multiplication de matrices

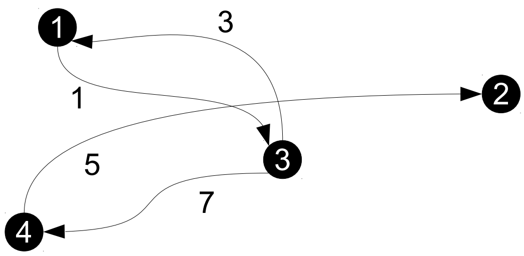
Exemple 1- On considère un graphe dont les sommets sont numérotés de *1* à *4*. Ce graphe est représenté par une matrice *N lignes-N colonnes* dans laquelle la valeur 1 placée en ligne *i* et colonne *j* indique l'**existence d'un arc** permettant de passer du point *i* au point *j*. Le graphe ainsi que les puissances de sa matrice de valuation[1] sont indiqués ci-dessous. Que remarquez-vous ?



$$G=\begin{pmatrix}0&0&1&0\\0&0&0&0\\1&0&0&1\\0&1&0&0\end{pmatrix}\qquad G^2=\begin{pmatrix}1&0&0&1\\0&0&0&0\\0&1&1&0\\0&0&0&0\end{pmatrix}$$

$$G^3=\begin{pmatrix}0&1&1&0\\0&0&0&0\\1&0&0&1\\0&0&0&0\end{pmatrix}\qquad G^4=\begin{pmatrix}1&0&0&1\\0&0&0&0\\0&1&1&0\\0&0&0&0\end{pmatrix}$$

A partir de l'exemple ci-dessous, que pensez-vous du cas des graphes dont les arcs sont valués (le coût associé à chaque arc peut-être différent de 1) ?

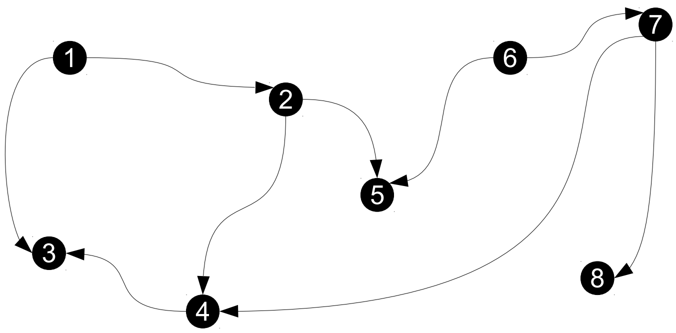


$$G=\begin{pmatrix}0&0&1&0\\0&0&0&0\\3&0&0&7\\0&5&0&0\end{pmatrix}\qquad G^2=\begin{pmatrix}3&0&0&7\\0&0&0&0\\0&35&3&0\\0&0&0&0\end{pmatrix}$$

Exemple 2- On considère de même un graphe dont les sommets sont numérotés de *1* à *8*. Ce graphe, sans cycle, et les puissances de sa matrice sont décrits ci-dessous. Que pensez-vous :

- des chemins existants dans ce graphe ?
- des chemins redondants et des plus courts chemins ?

1 La matrice de valuation ou d'adjacence, **G**, est telle que l'existence d'un arc entre un point source *s* et un point cible *c* est indiquée en *G[s][c]*.



$$G = \begin{pmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$G^2 = \begin{pmatrix} 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad G^3 = \begin{pmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad G^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Introduction aux recherches de plus courts chemins (PCC)

recherche de plus court chemin

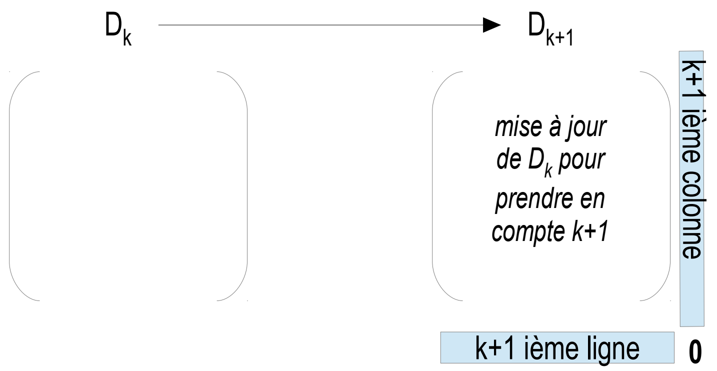
d'un sommet à tous les autres *à partir du sommet 1*	**entre tous les couples de sommets**
DIJKSTRA *(valuation ≥ 0)*	**DANTZIG** *(valuation quelconque)*
FORD-BELLMAN *(valuation quelconque)*	**FLOYD** *(valuation quelconque)*

M matrice d'adjacence avec dans M[i][j] le coût de l'arc ij
∞ si aucun arc entre i et j
0 si i=j

d tableau des distances avec dans d[i] distance entre 1 et i sur un PCC partant de1 ∞ sipas de PCC entre1 et i 0 si i=j	**D matrice des distances** avec dans D[i][j] distance entre i et j sur sur un PCC partant de i ∞ sipas de PCC entre1 et i 0 si i=j
p tableau de prédécesseurs avec dans p[i] prédécesseur de i sur un PCC partant de1 0 sipas de PCC entre1 et i	**P matrice de prédécesseurs** avec dans P[i][j] prédécesseur de j sur un PCC partant de i 0 sipas de PCC entre i et j

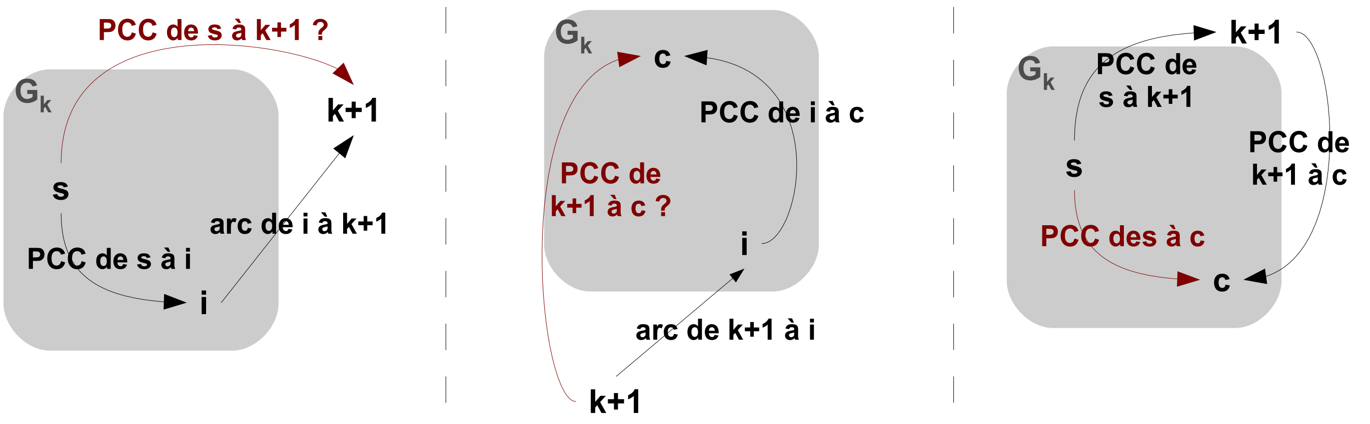
Algorithme de Dantzig

Cet algorithme est basé sur la construction pas à pas des PCC. On considère G_k le graphe réduit aux sommets d'indices 1 à k et D_k la matrice des distances associée. A chaque étape :



Cette mise à jour de D_k consiste en trois parties, illustrées ci-dessous, qui correspondent respectivement aux rôles possibles du sommet k+1 comme source, cible ou point intermédiaire dans les PCC considérés :

1) calcul des PCC entre les sommets s de G_k et le sommet $k+1$ (s est la source et ***k+1* est la cible des PCC**). Il s'agit du calcul de la $k+1^e$ colonne de D_k ;
2) calcul des PCC entre le sommet $k+1$ et les sommets c de G_k (***k+1* est la source des PCC** et c en est la cible). Il s'agit du calcul de la $k+1^e$ ligne ;
3) calcul des PCC entre deux sommets s et c de G_k dans lesquels ***k+1* est un sommet intermédiaire du PCC**. Il s'agit d'une mise à jour de la partie intérieure de la matrice G_k.



1) ajout de la colonne k+1 *2) ajout de la ligne k+1* *3) mise à jour des PCC sur les points 1 ..k*

Algorithme de Floyd

Cet algorithme travaille en construisant une suite de n matrices (de F_1 à F_n) à partir de la matrice d'adjacence notée F_0 dans laquelle on a le coût des arcs existants (0 pour la diagonale et $+\infty$ pour l'absence d'arc).

$F_k(x,y)$ est la longueur d'un PCC de x à y dont les sommets sont dans $1..k$ ($+\infty$ pour l'absence de PCC). Cette longueur est définie comme le minimum de $F_{k-1}(x,y)$ et de $F_{k-1}(x,k)+F_{k-1}(k,y)$.

Il faut vérifier à chaque étape l'existence d'une valeur F_k (x,x) négative qui révèle un cycle négatif et doit déclencher l'arrêt de l'algorihme.

Algorithme de Ford-Bellman

Cet algorithme consiste à minimiser aussi longtemps que possible le tableau des distances entre 1 et les autres sommets. Tant qu'il existe un arc tel que : *d[j] > d[i]+M[i][j]*, on met à jour le tableau des distances et celui des prédécesseurs.

L'arrêt peut se faire après un parcours de tous les arcs du graphe sans modification du tableau des distances.

Algorithme de Dijkstra

Cet algorithme ne peut être utilisé que si les valuations sont positives ou nulles

Il consiste à répartir les sommets en trois ensembles :

- *V* l'ensemble des sommets visités pour lesquels la distance à *1* (dans *d[]*) est connue. *V* est initialisé avec le point *1* (et *d[1]=0*) ;
- *A* l'ensemble des sommets atteints qui contient les successeurs immédiats non encore visités des éléments de *V*. Leur distance à *1* est estimée ;
- les autres sommets dont la distance à *1* est +∞.

A chaque étape, on choisit dans *A* le sommet qui a la plus petite distance estimée. Ce sommet est placé dans *V* avec toutes les mises à jour nécessaires. L'arrêt de l'algorithme se fait lorsque *V* contient tous les sommets du graphe.