Aix*Marseille université	Domaine Sciences et Technologies LICENCE INFORMATIQUE **Algorithmique 1 : TD 11** **Code UE : S04IN4A2**
Année 2019-2020	**Graphes**

Exercice 1 (Plus courts chemins et arcs de poids négatifs.)
(a) Soit G le graphe de sommets s, a, b, c, d, t, contenant les arcs $(s, a, 19)$, $(s, b, 2)$, $(a, t, 1)$, $(d, t, 7)$, $(c, d, -3)$, $(d, a, 2)$, $(c, a, -7)$, $(b, d, 10)$, $(a, b, 17)$, $(b, c, -2)$, $(c, t, 12)$. Appliquez l'algorithme de Bellman-Ford sur ce graphe en prenant s comme sommet source.

(b) On ajoute maintenant l'arc $(t, c, -9)$. Appliquez à nouveau l'algorithme de Bellman-Ford sur ce graphe. Que remarquez-vous ?

(c) Déduisez-en une approche pour déterminer si un graphe donné quelconque contient un circuit de poids négatif.

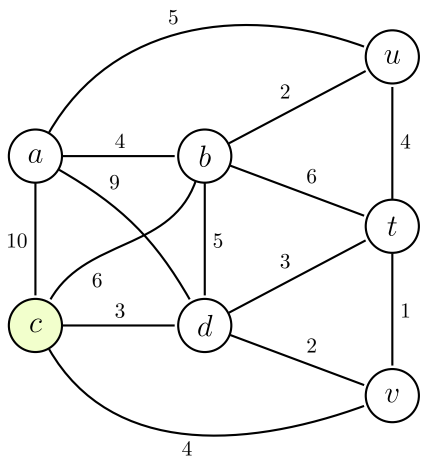
Exercice 2 (Plus courts chemins et arcs de poids négatifs.)
On considère un graphe $G = (S, A, l)$ qui contient des arcs de poids négatifs (mais pas de circuits de poids négatifs). Une approche efficace pour calculer les plus courts chemins entre tous les couples de sommets consiste, après l'application de l'algorithme de Bellman-Ford à partir d'un sommet s quelconque, à modifier les poids des arcs de façon à ce qu'il n'y ait plus d'arcs de poids négatif, puis à appliquer l'algorithme de Dijkstra (qui est très efficace) successivement à partir de chaque sommet. La modification des poids des arcs est la suivante : pour chaque arc $(u, v) \in A$, on remplace $l(u, v)$ par $l(u, v) + d(u) - d(v)$, où $d(u)$ et $d(v)$ désignent les longueurs des plus courts chemins produits par l'algorithme de Bellman-Ford entre s et u d'une part et s et v d'autre part. On note $G' = (S, A, l')$ le graphe obtenu par ce changement de pondération.

(a) Démontrez que dans G' aucun arc n'est de poids négatif.

(b) Démontrez que tout plus court chemin C dans G reste un plus court chemin dans G'.

(c) Evaluez le coût du calcul de tous les plus courts chemins (entre tous les couples de sommets du graphe) si on applique l'algorithme de Bellman-Ford successivement à partir de chacun des sommets du graphe. Que devient ce coût si on utilise l'algorithme de Dijkstra (avec un tas binaire) sur G'.

Exercice 3 (Arbres couvrants de poids minimal)



(a) Calculez un arbre couvrant de poids minimal dans le graphe ci-dessus en appliquant l'algorithme de Prim à partir du sommet c. Indiquez les valeurs des distances à chaque étape.

(b) Même chose en appliquant l'algorithme de Kruskal.