



**Figure 2.** Quelques exemples de graphes orientés

(si les deux sont présentes, cela revient à une liste d'adjacence).

Ces listes d'adjacence, ou listes de successeurs ou de prédécesseurs (ces dernières étant parfois, par abus de langage, désignées également par le terme de « liste d'adjacence »), sont généralement stockées dans un tableau ce qui permet de retrouver instantanément la liste d'adjacence du sommet  $x$

Ce que l'on nomme ici « liste » d'adjacence (ou de successeurs ou de prédécesseurs) est une suite de sommets, qui peut être réalisée algorithmiquement par un tableau, par une liste, ou par toute autre structure adaptée au stockage et à la recherche d'information ; pour ce qui nous intéresse ici nous coderons ces listes par des tableaux Java. Ainsi, le type qui représentera un graphe par listes d'adjacence sera le type `int[][]` mais ici ce type ne représente pas une matrice (dont toutes les lignes sont de même longueur) mais simplement une suite de tableaux qui ne sont pas tous astreints à être de même longueur.

Par exemple, si l'on prend le graphe 003 de la figure 1.b, sa matrice d'adjacence est une matrice à 13 lignes et 13 colonnes, et sa représentation par listes d'adjacence sera un