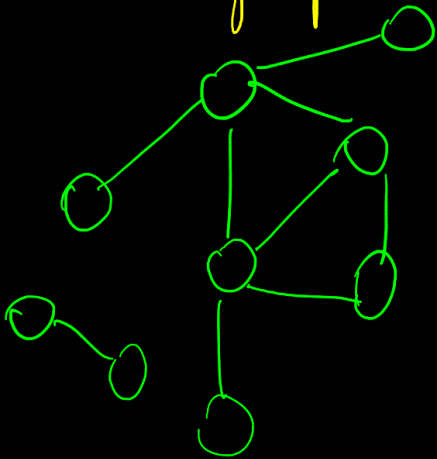


Un "patron" d'algo de graphe
à adapter pour résoudre plein
de problèmes!

Il vous faut

1. un graphe



2. un conteneur



3. de l'imagination.

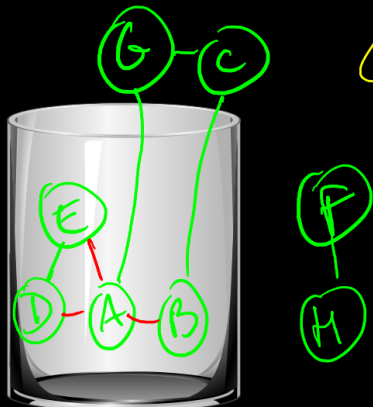
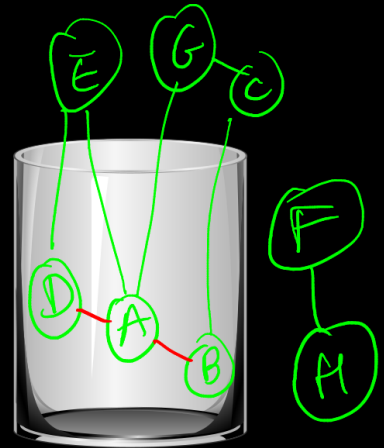
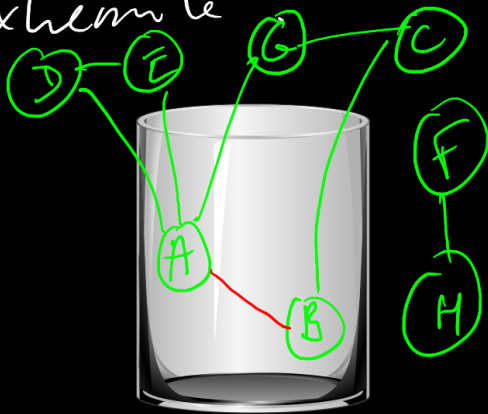
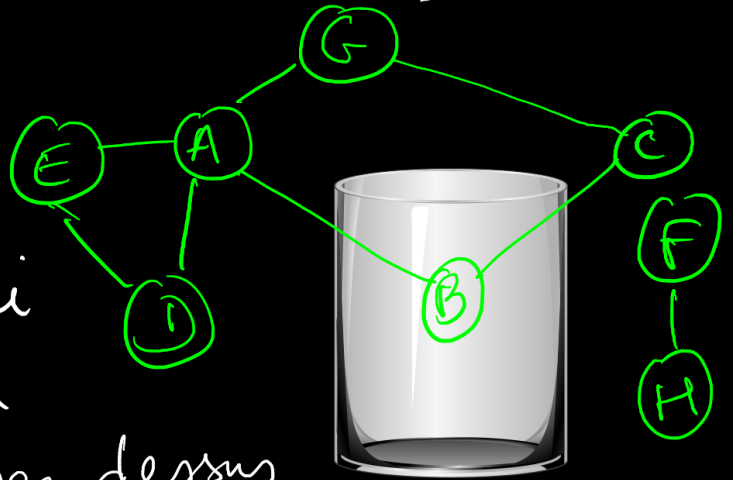
Au début

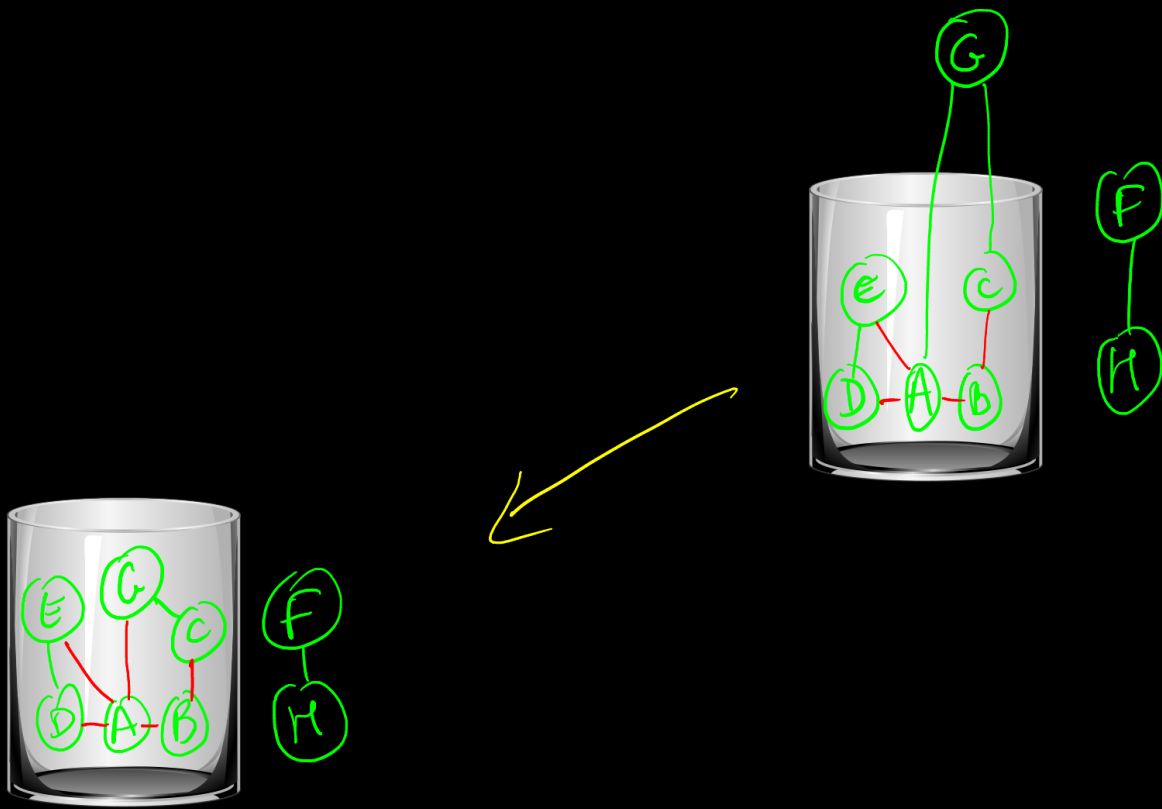
Selon les besoins

On met soit un sommet,
soit une arête (et ses extrémités)
dans le verre.

On continue

S'il y a des arêtes qui
sortent du verre, on
en choisit une pour tirer dessus
et la mettre dans le verre avec
son extrémité





Plus d'arête à cheval sur le vane.
 Possibilité d'introduire un vane supplémentaire
 pour traiter le reste.

Beaucoup d'algos suivent un schéma
 de ce type :

- décider si un graphe est connexe?
 (est-ce qu'un seul vane suffit)
- trouver un arbre couvrant dans un
 graphe connexe (la sélection d'arêtes
 rouges en est un)
- existe-t-il un chemin entre x et y ?
 si je commence avec x dans le vane, est-ce
 que j'arriverai à mettre y ?

- plus court chemin entre X et Y si les arêtes ont des poids positifs ?
choisir les arêtes à tirer de façon appropriée
- Depth First Search (DFS) et Breadth First Search (BFS) désignent des ordres particuliers pour tirer les arêtes.
- Décider si une arête est un "pont"
= retirer l'arête crée une composante de plus. On peut compter le nombre de vertex nécessaires avant et après.