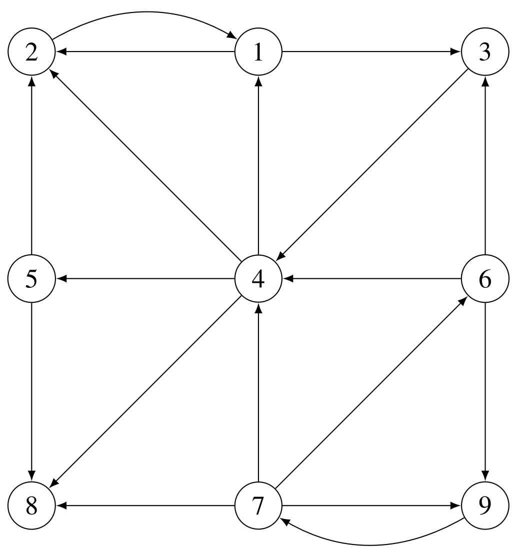
Université de Versailles - Saint Quentin

"Algorithmique de graphe"
TD2 : Parcours de graphes

1 Un premier parcours

Soit le joli graphe suivant :



Représentation : Les sommets sont considérés dans l'ordre croissant de leurs numéros et les successeurs de chaque sommet sont également classés par numéros croissants.

On considère par ailleurs l'algorithme suivant (les sommets étant numérotés de 1 à 9, les indices des tableaux font de même)

```
DEBUT
Pour chaque sommet x faire
  etat[x] = "non vu" ; pere[x] = Null ;
Fin Pour
i = 1 ; j = 1 ;
Pour chaque sommet x faire
  Si (etat[x] = "non vu") Alors
      atraiter[j] = x ; j = j+1 ; etat[x] = "vu"
      Tant que i < j faire
          y = atraiter[i] ; i = i+1 ;
          Pour chaque z successeur de y faire
              Si (etat[z] = "non vu") alors
                          etat[z] = "vu" ; atraiter[j] = z ; j = j+1 ; pere[z]= y ;
              Fin si
          Fin pour
      Fin tant que
  Fin si
Fin pour
FIN
```

1. Appliquer cet algorithme sur le graphe. Noter la (ou les) structure(s) arborescente(s) obtenue(s) lors du parcours des sommets.
2. Le tableau *atraiter* tient-il le rôle d'une file ou d'une pile ?
3. Quel est le parcours proposé par cet algorithme : profondeur ou largeur ?
4. Quelle est la raison d'être de la boucle
   Pour chaque sommet x faire - Si ($etat[x]$ = "non vu") ? Dans quel cas, cette boucle est t'elle inutile ?
5. Que donnerait ce parcours appliqué à partir du sommet 9 (les successeurs sont toujours classés par numéro croissant)
6. Quelle serait l'arborescence obtenue à partir de ce sommet 9 en effectuant l'autre type classique de parcours ?

2 Graphe biparti

Un graphe biparti est un graphe non orienté $G = (V, E)$ dont l'ensemble des sommets peut être partitionné en deux parties V_1 et V_2 telles que les aretes de E joignent toutes un sommet de V_1 et un sommet de V_2. On peut dire aussi qu'un graphe est biparti si son nombre chromatique est égal à 2 (il suffit de 2 couleurs pour colorier le graphe). On cherche un algorithme de reconnaissance de la famille des graphes bipartis. En entrée, on a donc un graphe, en sortie une réponse booléenne : oui s'il est biparti, non si ce n'est pas le cas.

1. Adapter l'algorithme de l'exercice précédent pour détecter si un graphe est biparti. Initialement, tous les sommets sont de couleur 0. Quand on traite un nouveau sommet, il prend la couleur 1. Les nouveaux voisins d'un sommet de couleur 1 sont mis à la couleur 2 (et vice-versa). Le tableau etat est donc remplacé par le tableau couleur.
2. Sur le graphe suivant (où sommets et voisins sont classés par numéros croissants), à quel moment détecte t'on que ce n'est pas un graphe biparti ?