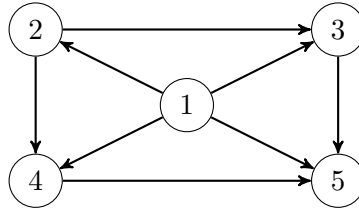


**Aufgabe 9.1** (4 Punkte)

Stelle den obigen Graphen durch

- ein Adjazenzarray
- Adjazenzlisten mit einfach zirkulär verketteten Listen
- eine Adjazenzmatrix
- verzeigerte Kantenobjekte

dar.

**Aufgabe 9.2** (4 Punkte) Betrachte folgenden Algorithmus, der prüft ob es in einem gerichteten Graphen  $G = (V, E)$  Kreise der Länge 2 gibt:

2CYCLE( $V, E$ )

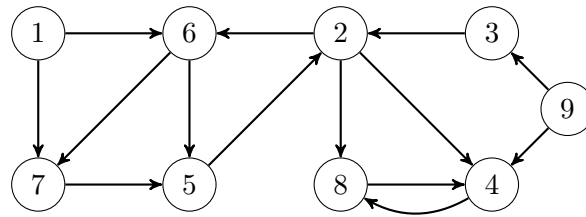
```
1 for each  $v \in V$ 
2     do for each outgoing edge  $(v, w) \in E$ 
3         do if  $(w, v) \in E$ 
4             then return TRUE
5 return FALSE
```

Der Algorithmus soll in-place arbeiten, also außer dem Eingabegraphen nur  $O(1)$  Speicherplatz benutzen. Gib die Laufzeit von diesem Algorithmus an, wenn  $G$  dargestellt ist als:

- eine ungeordnete Kantenfolge
- ein Adjazenzarray
- Adjazenzlisten
- eine Adjazenzmatrix

**Aufgabe 9.3** (4 Punkte)

Gegeben sei folgender Graph:



Durchlaufe den Graphen beginnend mit Knoten 1

- a) mit dem DFS-Verfahren
- b) mit dem BFS-Verfahren

wie in der Vorlesung beschrieben, so dass jeder Knoten höchstens einmal besucht wird. Gib dabei die Reihenfolge an, in welcher die Kanten besucht werden und die DFS-Nummern *dfsNum* und *finNum* für die Knoten. Gib auch für jede Kante ihren Typ (d.h. Baumkante, Vorwärtskante, Rückwärtskante, Querkante) an.

Falls von einem Knoten aus mehrere unbesuchte Knoten zu erreichen sind, wähle den Verweis, der auf den Knoten mit der kleinsten Zahl zeigt.

**Aufgabe 9.4** (4 Punkte)

Bei Tiefensuche in ungerichteten Graphen bietet es sich an, eine ungerichtete Kante  $\{v, w\}$  nur in einer Richtung zu inspizieren. Dies erreicht man dadurch, dass  $(w, v)$  in der Adjazenzliste von  $w$  „abgeschaltet“ wird, wenn  $(v, w)$  von  $v$  aus inspiziert wird. Zeige, dass die Tiefensuche in ungerichteten Graphen weder Vorwärts- noch Querkanten erzeugt.