Problem type 1:

Please answer the question below. Any algorithms given should be computationally efficient (please no brute-forcing). If needed, you may refer to the *Explore* algorithm (below) from lecture as a black box:

```
EXPLORE(G,u):
    Visited[1..n] ← false
    Add u to ToExplore and to S
    Visited[u] ← true
    While (ToExplore is non-empty)
        Remove node x from ToExplore
        for each edge xy in Adj(x)
            if (Visited[y] = false)
                Visited[y] ← true
                Add y to ToExplore
                Add y to S
    return S
```

(See variants below)

a. **BYG**

G is a directed graph and I want to know if node u can reach node v.

Solution:

- S=Explore(G,u)
- Check if $v \in S$

■

b. **BYD**

G is a directed graph and I want to find all nodes that u can reach.

Solution:

- S=Explore(G,u)

■

c. **BYA**

G is a directed graph and I want to find all nodes that can reach u .

Solution:

- Caluclate the reverse graph of G^{rev}
- $S = \text{Explore}(G^{rev}, u)$

■

d. **BYH**

G is a directed graph and I want to find all nodes in u's strong connected component.

Solution:

- Caluclate the reverse graph of G^{rev}
- $\textsc{Explore}(G, u) \cap \textsc{Explore}(G^{rev}, u)$

■

Problem type 2:

Answer the following problem:

(See variants below)

a. **BYC**

What type of *directed*-graph has only one strongly connected component?

Solution: Two solutions I'd accept:

- A fully connected graph.
- A graph that is one large cycle

■

b. **BYF**

Assuming a directed graph with n nodes, how many edges do you need to make the graph have a single strongly connected component.

Solution: A directed graph that has the minimal number of edges but is still strongly connected is a cycle. So n edges. (Technically if $n = 1$ then you don't need any edges. Thank you TA Owen for pointing this out) ■

c. **BYE**

Assuming a *undirected* graph with n nodes, how many edges do you need to make the graph connected.

Solution: You'd need $n - 1$ edges to connect all the nodes in a undirected graph together. ■

d. **BYB**

Assuming a *directed* graph with n nodes, what type of graph would have the most number of cycles of size n. How many cycles would this graph have.

Solution: Assuming a fully connected graph, you can construct $n!$ cycles of size n (you have n choices for the first node, $n-1$ for the second node in the cycle, etc.). ■