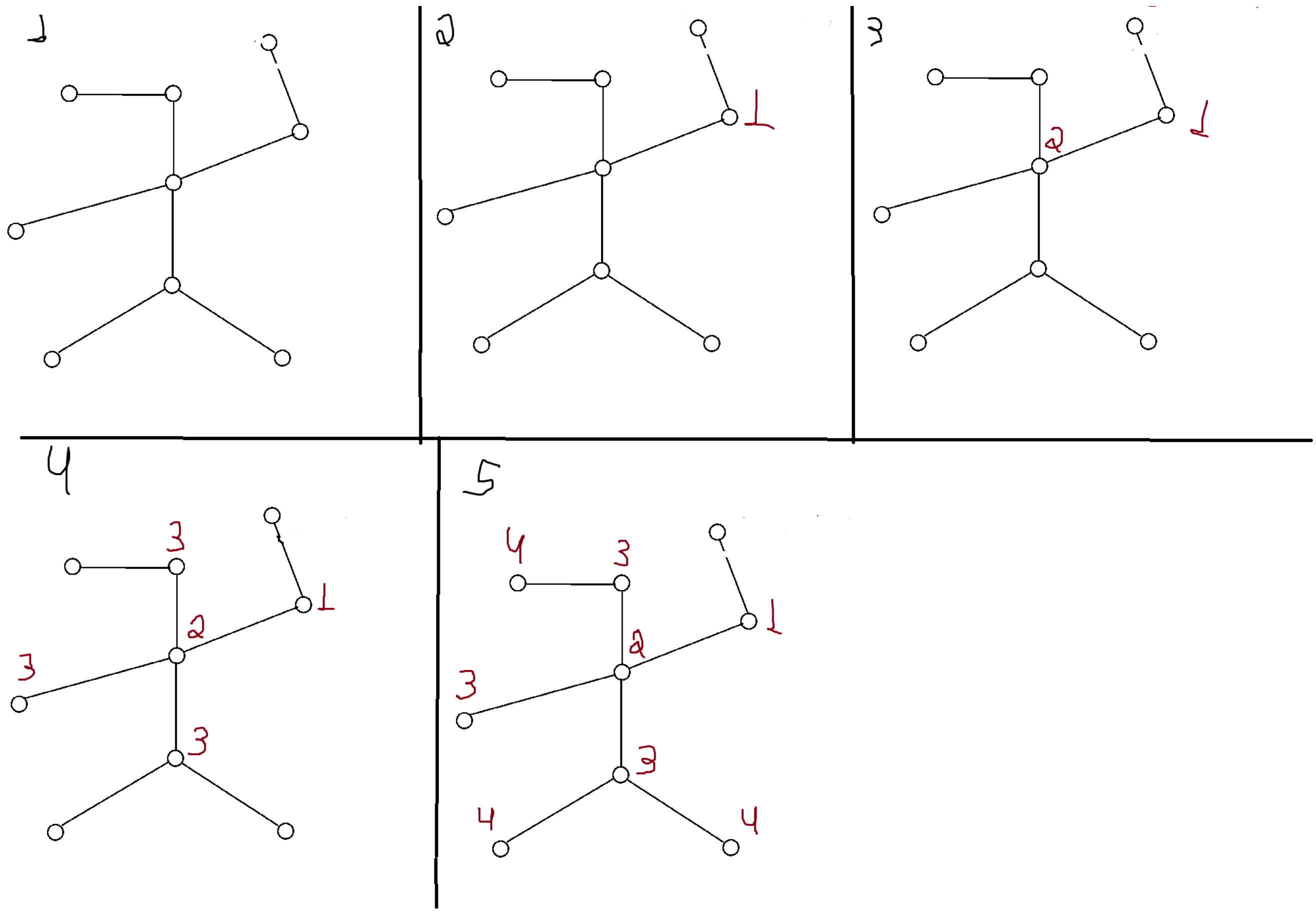
Questão 1

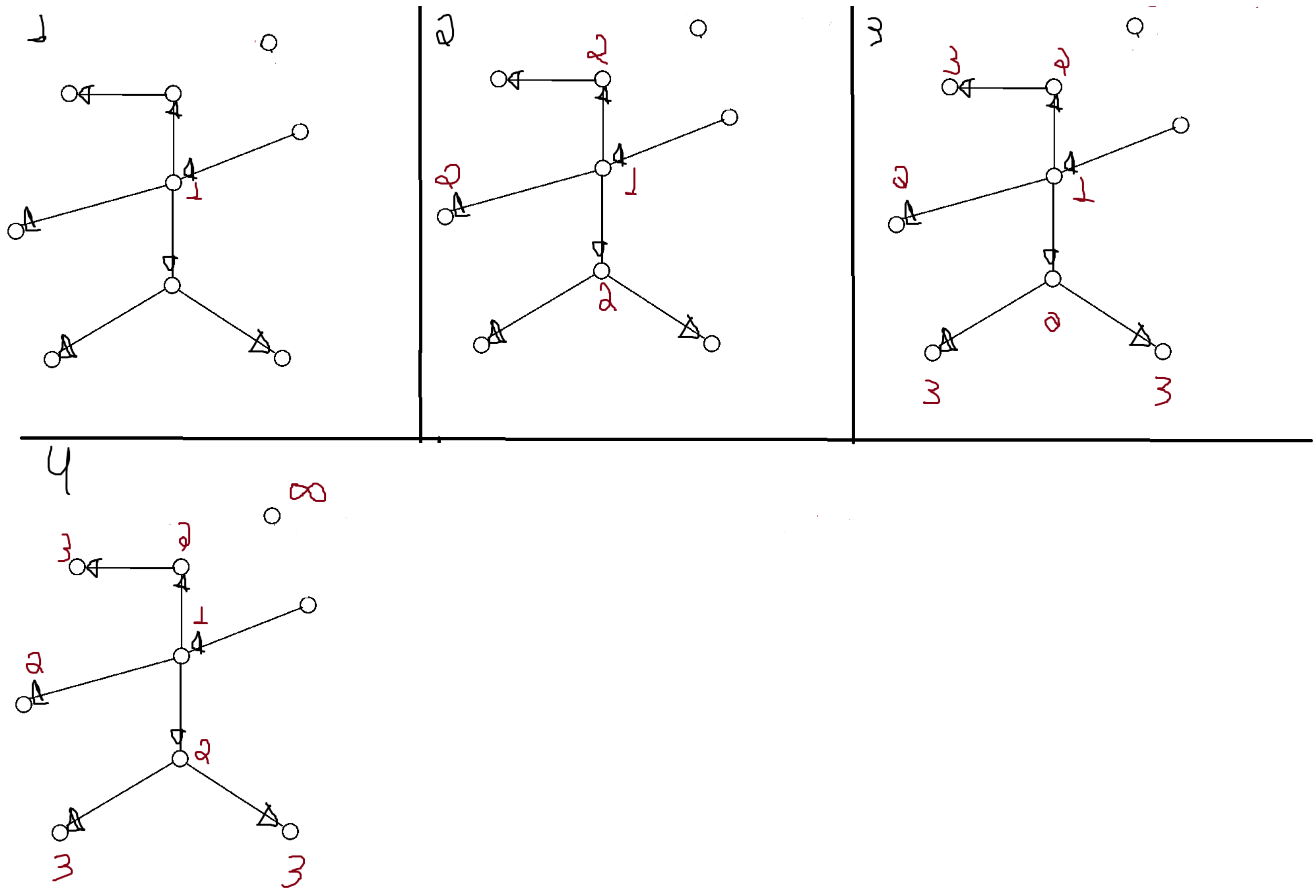
a) O diâmetro é a maior de todas as excentricidades do grafo, portanto calculando a excentricidade de todos os vértices do grafo e pegando o maior deles será o diâmetro. Para calcular a excentricidade utiliza-se o algoritmo de busca em largura (BFS) modificado. O algoritmo é o seguinte:

```
diametro( G ):
    X = \[\]
    para cada $v \in V$:
        X.append( excentricidade( v, G ) )
    retornar a maior excentricidade

excentricidade( v, G ):
    fila = \[\]
    d = \[\]
    d\[v\] = 0
    insere v na fila
    marca v como visitado
    enquanto a fila não estiver vazia
        v = remove da fila
        para cada u vizinho de v
            se u não foi visitado
                d\[u\] = d\[v\] + 1
                marca u como visitado
                insere u na fila
    retornar a maior distância
```



b) O algoritmo para calcular o diâmetro em um grafo direcionado e fortemente conexo é o mesmo na questão anterior. O ponto em questão é o fato de não ser fortemente conexo, pois nessa situação pode haver um vértice que não é atingido por nenhum outro vértice, isto é, o fecho transitivo inversos desse vértice é ele mesmo e nesse caso o diâmetro encontrado pelo algoritmo será infinito.



Questão 3

- Para encontrar uma árvore $T = (V, E')$ garantindo que uma aresta específica $e \in E$ esteja presente em T e que essa árvore seja mínima:

```
1. modificarPesoAresta( e, G )
    1. peso = acharMenorPeso( G )
    2. peso = peso -1
    3. w(e) = peso
2. prim( G )
```

Com o algoritmo acima, modificamos o peso da aresta escolhida para o menor peso do grafo menos 1, assim garantindo que a aresta escolhida seja a menor, então aplicar o algoritmo de Prim, após isso restaurar o peso da aresta para o peso original.

OU

```
w(e) = - infinito
```

É mais direto, e mais computacionalmente viável (segundo o Gualtiere).

- Para que $F = (V, E' \setminus \{e\})$ seja uma floresta geradora mínima de $G = (V, E' \setminus \{e\})$:

Nesse caso, a aresta e escolhida não estará presente no grafo nem na floresta geradora mínima. Assim, uma solução seria executar o algoritmo de Prim, mas com o grafo com a aresta escolhida já removida, isto é: $\text{prim}(G = (V, E' \setminus \{e\}))$.

Questão 4

Sim, é possível identificar um grafo acíclico utilizando o algoritmo para encontrar componentes conexos, pois o algoritmo aplica a primeira DFS para encontrar o vértice inicial e aplica a segunda DFS no grafo transposto para encontrar a quantidade de componentes conexos a partir da quantidade de rótulos diferentes dos vértices, se todos os vértices possuírem rótulos diferentes entre si tal que $|CFC| = |V|$, isto é a quantidade de componentes fortemente conexos é igual a quantidade de vértices do grafo, então o grafo é acíclico.