Question from a to c

```
Question a:
[[0, 1, 1, 0], [0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 0]]
The graph is a DAG
[[0, 1, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [0, 0, 1, 0]]
The graph is not a DAG
Question b:
[[0, 1, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [0, 0, 1, 0]]
The graph is not a DAG
The new dag is : [[0, 1, 0, 0], [0, 0, 0, 1], [0, 0, 0, 0], [0, 0, 1, 0]]
Question c:
The equivalent sorted graph is:[[0, 1, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [0, 0, 1, 0]]
```

Question d

H does not equal to G

The sort_dag function takes a directed acyclic graph (DAG) as input and returns a topologically sorted version of the same graph. This means that the nodes in the sorted graph will be ordered in such a way that for every directed edge u -> v, node u will appear before node v in the sorted list which may not be a DAG. On the other hand, the make_dag function returns a DAG representation of the same graph. Therefore, G does not equal to H.