① @ Here we created a matrix (empty) and as we found a u, and a v, we just put them in their proper coordinating position. We get the u, v values by slicing them using split and iterating them by the first two numbers. Here we used slicing ~~utie~~ also while finding and seperating their weights.

① (b) This time we also create a matrix but having empty lists as values and whenever we found 3 values we splitted them and kept them in their corresponding position in a tuple form and added other tuples in the same list if there was another 2 values for that position.

② Here we started at the node 1 and checked its neigbor/adjacent nodes and went on to find their adjacent nodes when they are connected and not visited and when all the nodes are visited, we returned all the nodes visitable from starting node.

③ Here we also did something Similar to bfs but we used the principles of stack for this (Last in first out) when popping from the queue instead of Fifo like bfs. Here we keep seaching untill we find a node with no adjacent and tay poping others untill we have no unvisited node left.

④ Here we made a function that finds if there is a way to come back to the starting node and ran it on every single node of the graph. whenever we even find one cycle, on a way of coming bac to the nodes we start from we stop and return flags^ (True if found) that then determine if we have cycle on not.

⑤ Here we used a similar method to bfs where we found out the paths connecting nodes and then we traversed neigbors as we were on search for destination when we search for destination and find it we save the path and show the result. As it is a positive graph we had to check only onee. We used the stack's lifo method here too.

⑥ Here we started from a node, collected diamonds as we moved around using adjacent nodes that were in range movable to and whenever we got a diamond we added it to final count and ran the same process on adjacent nodes finally adding them and comparing the total amounts found each run and outputing the maximum diamond count we got in the end.