It creates a list called visit, which will store the visited status of each city. It creates a list called ad-l, which stores the adjacency list of the graph. It loops through next e lines of the input files, each containing two integers, f and t. The code appends t to the list at index f in the ad_l list. cycle_detection takes two parameters ad_l, select. The func performs DFS traversal, starting from the selected city and returns True if it finds a cycle, otherwise False. It marks the selected city as visited by setting visit[select] to 1. It loops through the list at index select in the ad_l list, which contains cities of select. For each adjacent ~~list~~ city ad_nd, it checks if it is not visited by comparing visit[ad_nd] to 0. If so, it recursively calls the cycle detection function and assigns the return value to the variable Crot_cycle. It

Got-cycle is True, it return true, indicating a cycle is found. If the adjacent city ad-nd is already visited by comparing visit[ad-nd] to 1, it returns True. Else it returns False.