Task 1:

First of all, we will keep track of parents node and friend circle sizes. Initialize the friends array with each node have one friend. Then iterate through each friendship connection and check if the root parents nodes are not same, then update the parents node and also the size of friend circle.

Task 2:

Firstly, initializes a array to track parents node and sort connection by the cost. then iterates through each connection and check if the root parent nodes of the connected nodes by recursive way. If the root parent nodes are different, it updates the parents node and adds the connection to the graph with calculating to total cost.

Task 3:

Firstly, we will create a dictionary where the keys are fibonacci values and values are indexes. Now, recursively we will check if the value of number is already in storage if not then it goes again using recursive way, the base case will be if number = 0/1 we will return 0/1 other wise will do again.

Task 4:

By using, BFS, approach, firstly make a queue, and visited set, and poping from the queue and explores possible sum of coins by incrementing the number of steps for each iteration. Recursively, we will do it untill we explore all possible sums are not explored.