

Task 1:

After taking the inputs correctly sorted the list of roads based on their cost. X variable will keep track of connected components. Initializes $s=0$ to store minimum weight. Inside the loop extracted a,b,c respectively then initializes $m=n=-1$ to store indices of the connected components. In the nested loop if value a is inside x then storing it's indices in m, same for b. Then check m and n are equal or not. If not equal it means adding them will not create any cycle. After adding them deleting their individual value. Lastly appending the adding value inside x, and incrementing s.

Task 2:

After taking inputs create an empty dictionary to store subproblems value(for memorization) called memo. Created the find function, if the desired value in memo then returning it or the value is 0 or 1 then returning 1, it is actually base case of this function. Then calling the find function like Fibonacci number, then adding the value in memo dictionary lastly returning it. It is actually recursive function.

Task 3:

Initialize a list min_coins of length $x + 1$ where each element represents the minimum number of coins needed to make up the corresponding amount. Initially, set all elements to infinity. The first loop iterates through every coins. The inner loop checks if adding the current coin to a smaller amount and incrementing the coin count by 1 results in a smaller value for the current amount. If true, update min_coins with the new minimum value. If the number can't be made using that coin then it will return -1. And lastly if min_coins[x] is not infinity then it would be the correct output.