

Task 1: This Python script reads input from a file named "input1.txt" and writes output to a file named "output1.txt". It implements a disjoint set union (DSU) data structure to efficiently handle union-find operations. Specifically, it defines functions `find_representative` to find the representative of a set and `union` to merge two sets together, updating the representative and circle size accordingly. Finally, it closes both input and output files after processing.

Task 2: This Python script reads input from a file named "input2.txt" and writes output to a file named "output2.txt". It constructs an adjacency list representation of a graph with cities and roads, then applies Prim's algorithm to find the minimum cost of spanning the graph, starting from city 1. Finally, it prints the minimum cost to the output file and closes both input and output files after processing.

Task 3: This Python script reads input from a file named "input3.txt" and writes output to a file named "output3.txt". It calculates the number of ways to climb stairs (represented by the integer 'freddy') using a recursive approach optimized with memoization. Finally, it writes the result to the output file and closes both input and output files after processing.

Task 4: This Python script reads input from a file named "input4.txt" and writes output to a file named "output4.txt". It implements a recursive function to find the minimum number of coins required to make a target value using a given set of coin denominations. The function `'Min_Coin_Change'` recursively explores all possible combinations, choosing the minimum among them. If no valid combination is found, it writes "-1" to the output file. Finally, it closes both input and output files after processing.