Beaumont Yin
CS 146

Task: There are n houses in a village. We want to supply water for all the houses by building wells and laying pipes.

For each house i, we can either build a well inside it directly with cost wells[i - 1] (note the -1 due to 0-indexing), or pipe in water from another well to it. The costs to lay pipes between houses are given by the array pipes where each pipes[j] = [house1j, house2j, costj] represents the cost to connect house1j and house2j together using a pipe. Connections are bidirectional, and there could be multiple valid connections between the same two houses with different costs.

Return the minimum total cost to supply water to all houses.

Approach:

1. Create a list to store the edges
2. Add virtual node connections for houses
3. For each pipe add connections between each house
4. Sort the edges based on their weight
5. Create an array and function for union-find
6. Use Kruskal's algorithm to find the Minimum Spanning Tree
7. Return the sum of the edges of the MST

Test cases:

```
// Test Case 1
    int n1 = 2;
    int[] wells1 = {1, 1};
    int[][] pipes1 = {{1, 2, 1}, {1, 2, 2}};
    int output1 = solution.minCostToSupplyWater(n1, wells1, pipes1);
    System.out.println("Test Case 1: " + (output1)); //should be 2

    // Test Case 2
    int n2 = 3;
    int[] wells2 = {1, 2, 2};
    int[][] pipes2 = {{1, 2, 1}, {2, 3, 1}};
    int output2 = solution.minCostToSupplyWater(n2, wells2, pipes2);
    System.out.println("Test Case 2: " + (output2));//should be 3
```