Editorial-Minimal Cost

Here,they are given a set of cities(numbered from ***1*** to ***N***) that are connected with directed roads. Each city has a value assigned to it.This is clearly a graph problem
represent vertices and edges represent roads. The goal is to find a path in the graph that will have the minimum GCD,when the values of the vertices are considered.

We can use the Euclidean algorithm for finding GCD. You have to focus on the fact that the GCD of two numbers cannot be larger than any of the numbers itself.

If we were to calculate the minimum gcd of a set of numbers; We can compute the gcd of two numbers and take the result as the minimum GCD.Thereafter we can compute the compute the gcd of current minimum GCD and another GCD which will give us the minimum GCD at end due to the reason above.What we have to do is take all the paths in the graph and compute min gcd for values in each path.We can use dynamic programming to reduce redundant computation.

Algorithm

First, take inputs and create an adjacency list representation of the graph.Then, start a DFS traversal of the graph from each node and calculate the gcd and store minimum.

How DFS traversal to find minimum GCD works

```
take min_gcd as input (this is the value of the node at start)
minimum = Infinity

for each neighbor "N" of current Vertex  :
    if N is already visited goto to next neighbor;

    compute the gcd of min_gcd and N ; assign it back to min_gcd
    minimum = min(min_gcd, minimum);

    Mark N as visited

    dfs_result = call dfs starting from N with min_gcd and visited array
    assign min( dfs_result , minimum)  back to minimum

return minimum;
```

Use Dynamic Programming to improve performance and store calls to dfs and return if there is result that has been already computed.