990. Satisfiability of Equality Equations

You are given an array of strings `equations` that represent relationships between variables where each string `equations[i]` is of length `4` and takes one of two different forms: `"xi==yi"` or `"xi!=yi"`.Here, `xi` and `yi` are lowercase letters (not necessarily different) that represent one-letter variable names.

Return `true`* if it is possible to assign integers to variable names so as to satisfy all the given equations, or `false` otherwise*.

Example 1:

```
Input: equations = ["a==b","b!=a"]
Output: false
Explanation: If we assign say, a = 1 and b = 1, then the first equation is satisfied, but not the second.
There is no way to assign the variables to satisfy both equations.
```

Example 2:

```
Input: equations = ["b==a","a==b"]
Output: true
Explanation: We could assign a = 1 and b = 1 to satisfy both equations.
```

Example 3:

```
Input: equations = ["a==b","b==c","a==c"]
Output: true
```

Example 4:

```
Input: equations = ["a==b","b!=c","c==a"]
Output: false
```

Example 5:

```
Input: equations = ["c==c","b==d","x!=z"]
Output: true
```

```python
class Solution:
    def equationsPossible(self, equations: List[str]) -> bool:
        parent = [i for i in range(26)]
        rank = [1]*26

        for ele in equations:
            if ele[1]=='=':
                self.union(ord(ele[0])-97,ord(ele[3])-97,rank,parent)

        for el in equations:
            if el[1]=='!':
                lx = self.find(parent,ord(el[0])-97)
                ly = self.find(parent,ord(el[3])-97)
                if lx==ly:
                    return False
        return True

    def find(self,parent,x):
        if parent[x]==x:
            return x
        temp = self.find(parent,parent[x])
        parent[x] = temp
        return temp

    def union(self,x,y,rank,parent):
        lx = self.find(parent,x)
        ly = self.find(parent,y)

        if lx!=ly:
            if rank[lx]>rank[ly]:
                parent[ly] = lx
            elif rank[lx]<rank[ly]:
                parent[lx]=ly
            else:
                parent[ly] = lx
                rank[lx] = rank[lx]+1
```