21.you are given a string s, and an array of pairs of indices in the string pairs where pair[i]=[a,b] indicates 2 indices of the string you can swap the character at any pair of indices in the given pairs any numbers of times return the lexicographically smallest string that s can be changed to after using the swaps.

Aim: utilize the union find data structure to group indices that can be swapped, sort character within each group and returnstruct

Program: def smallestStringWithSwaps(s, pairs):

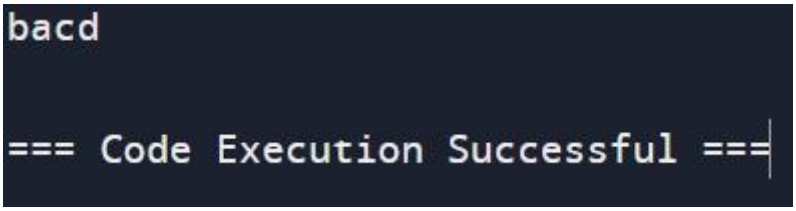
```
    def find(x):
        if x != parent[x]:
            parent[x] = find(parent[x])
        return parent[x]
    def union(x, y):
        root_x, root_y = find(x), find(y)
        if rank[root_x] > rank[root_y]:
            parent[root_y] = root_x
        else:
            parent[root_x] = root_y
            if rank[root_x] == rank[root_y]:
                rank[root_y] += 1
    parent = list(range(len(s)))
    rank = [0] * len(s)
    for pair in pairs:
        union(pair[0], pair[1])
    groups = collections.defaultdict(list)
    for i in range(len(s)):
        groups[find(i)].append(s[i])
    for group in groups:
        groups[group].sort(reverse=True)
    res = []
    for i in range(len(s)):
        res.append(groups[find(i)].pop())
    return ''.join(res)
```

s = "dcab"

pairs = [[0, 3], [1, 2]]

print(smallestStringWithSwaps(s, pairs))

Output:



time complexity:O(n+m(logm))