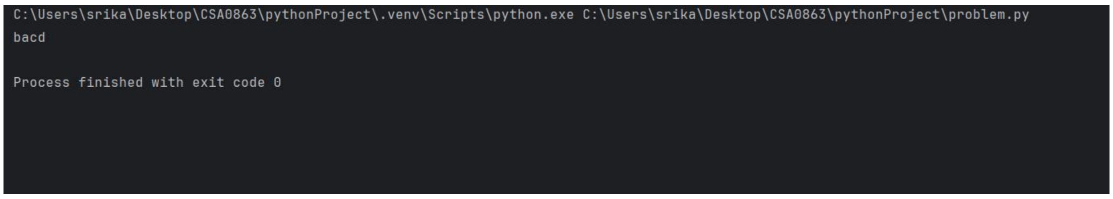
Q)You are given a string s, and an array of pairs of indices in the string pairs where pairs[i] = [a, b] indicates 2 indices(0-indexed) of the string.You can swap the characters at any pair of indices in the given pairs any number of times. Return the lexicographically smallest string that s can be changed to after using the swaps.

Program:

```
def smallestStringWithSwaps(s, pairs):
    from collections import defaultdict, deque
    graph = defaultdict(list)
    for a, b in pairs:
        graph[a].append(b)
        graph[b].append(a)
    def find_connected_component(node, visited,
component):
        stack = [node]
        while stack:
            current = stack.pop()
            if current not in visited:
                visited.add(current)
                component.append(current)
                for neighbor in graph[current]:
                    if neighbor not in visited:
                        stack.append(neighbor)
    visited = set()
    components = []
    for i in range(len(s)):
        if i not in visited:
            component = []
            find_connected_component(i, visited,
component)
            components.append(component)
    s = list(s)
    for component in components:
        indices = sorted(component)
        chars = sorted(s[i] for i in indices)
        for index, char in zip(indices, chars):
            s[index] = char
    return ''.join(s)
s = "dcab"
pairs = [[0, 3], [1, 2]]
print(smallestStringWithSwaps(s, pairs))
#  Output: "bacd"
```

Output:



```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe C:\Users\srika\Desktop\CSA0863\pythonProject\problem.py
bacd

Process finished with exit code 0
```

Time complexity:O(nlogn)