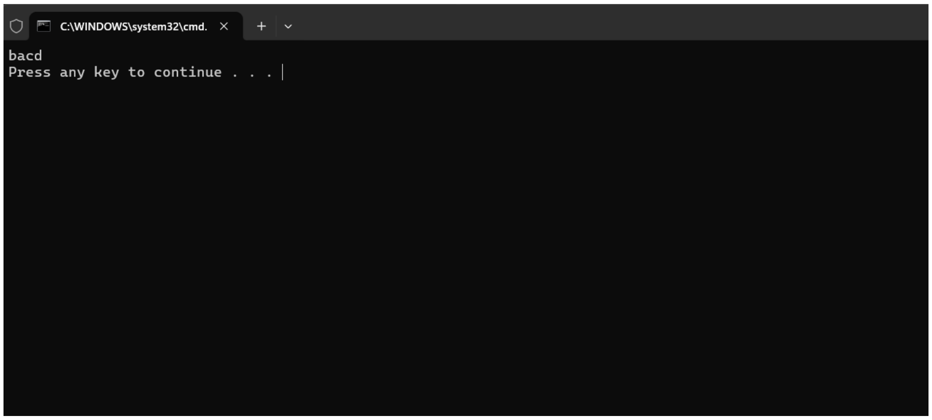
30) You are given a string s, and an array of pairs of indices in the string pairs where pairs[i] = [a, b] indicates 2 indices(0-indexed) of the string.You can swap the characters at any pair of indices in the given pairs any number of times. Return the lexicographically smallest string that s can be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

CODE:

```
import collections


def smallestStringWithSwaps(s, pairs):
    def find(x):
        if parent[x] != x:
            parent[x] = find(parent[x])
        return parent[x]

    def union(x, y):
        root_x, root_y = find(x), find(y)
        if root_x != root_y:
            parent[root_x] = root_y

    parent = {i: i for i in range(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)

    result = []
    for i in range(len(s)):
        result.append(groups[find(i)].pop())

    return ''.join(result)
s = "dcab"
pairs = [[0, 3], [1, 2]]
print(smallestStringWithSwaps(s, pairs))
```

OUTPUT:



```
bacd
Press any key to continue . . .
```

TIME COMPLEXITY : O(nlogn)