1. Lexicographically Smallest String After Swaps

```
python
Copy code
def smallestStringWithSwaps(s, pairs):
    from collections import defaultdict
    parent = list(range(len(s)))

    def find(x):
        if parent[x] != x:
            parent[x] = find(parent[x])
        return parent[x]

    def union(x, y):
        rootX, rootY = find(x), find(y)
        if rootX != rootY:
            parent[rootY] = rootX

    for a, b in pairs:
        union(a, b)

    components = defaultdict(list)
    for i in range(len(s)):
        components[find(i)].append(i)

    res = list(s)
    for indices in components.values():
        indices.sort()
        chars = sorted(res[i] for i in indices)
        for i, char in zip(indices, chars):
            res[i] = char

    return ''.join(res)

# Example usage
s = "dcab"
pairs = [[0, 3], [1, 2], [0, 2]]
print(smallestStringWithSwaps(s, pairs))  # Output: "abcd"
```

2. Check If One String Can Break Another

```
python
Copy code
def checkIfCanBreak(s1, s2):
    s1, s2 = sorted(s1), sorted(s2)
    return all(a >= b for a, b in zip(s1, s2)) or all(a <= b for a, b in
zip(s1, s2))

# Example usage
s1 = "abc"
s2 = "xya"
print(checkIfCanBreak(s1, s2))  # Output: True
```

3. Minimize Value by Replacing '?'

```
python
Copy code
def minimizeStringValue(s):
    res = []
    for c in s:
        if c == '?':
            for replacement in 'abcdefghijklmnopqrstuvwxyz':
                if (not res or res[-1] != replacement):
                    res.append(replacement)
                    break
        else:
            res.append(c)
    return ''.join(res)

# Example usage
s = "a?c"
print(minimizeStringValue(s))  # Output: "abc"
```

4. String Before Last Removal

```
python
Copy code
def stringBeforeLastRemoval(s):
    from collections import defaultdict
    import string

    while True:
        seen = set()
        for char in string.ascii_lowercase:
            pos = s.find(char)
            if pos != -1:
                seen.add(char)
                s = s[:pos] + s[pos+1:]
        if not seen:
            break
    return s

# Example usage
s = "aabcbbca"
print(stringBeforeLastRemoval(s))  # Output: "ba"
```

5. Subarray with Largest Sum

```
python
Copy code
def maxSubArray(nums):
    current_sum = max_sum = nums[0]
    for num in nums[1:]:
        current_sum = max(num, current_sum + num)
        max_sum = max(max_sum, current_sum)
    return max_sum

# Example usage
nums = [-2,1,-3,4,-1,2,1,-5,4]
print(maxSubArray(nums))  # Output: 6
```

6. Maximum Binary Tree

```
python
Copy code
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def constructMaximumBinaryTree(nums):
    if not nums:
        return None
    max_index = nums.index(max(nums))
    root = TreeNode(nums[max_index])
    root.left = constructMaximumBinaryTree(nums[:max_index])
    root.right = constructMaximumBinaryTree(nums[max_index + 1:])
    return root

# Example usage
nums = [3,2,1,6,0,5]
root = constructMaximumBinaryTree(nums)  # The root of the maximum binary
tree
```

7. Maximum Sum of Circular Subarray

```
python
Copy code
def maxSubarraySumCircular(nums):
    def kadane(gen):
        current_sum = max_sum = next(gen)
        for num in gen:
            current_sum = max(num, current_sum + num)
            max_sum = max(max_sum, current_sum)
        return max_sum

    total_sum = sum(nums)
    max_kadane = kadane(iter(nums))
    min_kadane = kadane(-num for num in nums)
    return max(max_kadane, total_sum + min_kadane)

# Example usage
nums = [1,-2,3,-2]
print(maxSubarraySumCircular(nums))  # Output: 3
```

8. Maximum Sum of Non-Adjacent Subsequence

```
python
Copy code
def maxSumAfterQueries(nums, queries):
    mod = 10**9 + 7
    def max_non_adjacent_sum(arr):
        incl = excl = 0
        for num in arr:
            incl, excl = max(excl + num, incl), incl
        return max(incl, excl)

    total_sum = 0
    for pos, x in queries:
        nums[pos] = x
        total_sum = (total_sum + max_non_adjacent_sum(nums)) % mod

    return total_sum

# Example usage
nums = [1, 2, 3]
```

```
queries = [[1, 5], [0, 4]]
print(maxSumAfterQueries(nums, queries))  # Output: 14
```

9. k Closest Points to Origin

```
python
Copy code
import heapq
def kClosest(points, k):
    return heapq.nsmallest(k, points, key=lambda p: p[0]**2 + p[1]**2)

# Example usage
points = [[3,3],[5,-1],[-2,4]]
k = 2
print(kClosest(points, k))  # Output: [[3,3],[-2,4]]
```

10. Median of Two Sorted Arrays

```
python
Copy code
def findMedianSortedArrays(nums1, nums2):
    nums = sorted(nums1 + nums2)
    mid = len(nums) // 2
    if len(nums) % 2 == 0:
        return (nums[mid - 1] + nums[mid]) / 2
    else:
        return nums[mid]

# Example usage
nums1 = [1, 3]
nums2 = [2]
print(findMedianSortedArrays(nums1, nums2))
```