DSA Lab assignment 02

Problem 01

You are given a stack of N integers such that the first element represents the top of the stack and the last element represents the bottom of the stack.

You need to pop at least one element from the stack. At any one moment, you can convert stack into a queue. The bottom of the stack represents the front of the queue.

You cannot convert the queue back into a stack. Your task is to remove exactly K elements such that the sum of the K removed elements is maximised.

Input format :

```
The first line consists of two space-separated integers N and K.
The second line consists of N space-separated integers denoting the elements of the stack.
```

Output format :

```
Print the maximum possible sum of the K removed elements
```

Constraints:

```
1 <= N <= 10^5
1 <= K <= N
1 <= Ai <= 10^9
```

1st Method

```
n , k = map(int,input().split())             # stack =  the stack itself
stack = list(map(int,input().split()))          # n = the number of elements in the stack
                                                # k = the number of elements to be removed
s = 0
removed = []
for i in range(k):
    s += stack[i]                           # s = keeping the track of sum of the first k elements in the stack
    removed.append(stack[i])

m = s
print(m)
for i in range(k-1):
    s = s - stack[k-i-1] + stack[n-1-i]
    removed.append(stack[k-i-1])
    if s >= m:
        m = s

print("Removed elements: ", removed)
print("Sum of removed elements:", m)
```

```
10 5
10 9 1 2 3 4 5 6 7 8
25
Removed elements:  [10, 9, 1, 2, 3, 3, 2, 1, 9]
Sum of removed elements: 40
```

2nd Method

```
def maximum_sum(n, k, stack):
    stack.sort(reverse=True)      # stack is sorted in descending order using the sort method
    removed = stack[:k]              # first k elements are removed
    return sum(removed), removed

n, k = map(int, input().split())              # --Three arguments--
stack = list(map(int, input().split()))      # n = the number of elements in the stack, stack =  the stack itself (stack)
sum_of_removed, removed = maximum_sum(n, k, stack)
print("Sum of removed elements:", sum_of_removed)
print("Removed elements:", removed)

```

```
10 5
10 9 1 2 3 4 5 6 7 8
Sum of removed elements: 40
Removed elements: [10, 9, 8, 7, 6]
```

Problem 02

You are given two arrays each of size n, a and b consisting of the first n positive integers each exactly once, that is, they are permutations.

Your task is to find the minimum time required to make both the arrays empty. The following two types of operations can be performed any number of times each taking 1 second:

In the first operation, you are allowed to rotate the first array clockwise. In the second operation, when the first element of both the arrays is the same, they are removed from both the arrays and the process continues.

Input format

```
The first line contains an integer n, denoting the size of the array a.
The second line contains the elements of array b.
The third line contains the elements of array
```

. **Output format**

```
Print the total time taken required to empty both the array.
```

Constraints

```
1 <= n <= 100
```

```
def minimum_time(n, a, b):        # takes in the size of the arrays (n) and the arrays themselves (a and b)
    time = 0
    i = 0
    while len(a) > 0 and len(b) > 0:     # The loop continues until either a or b is emptied.
        while a[0] != b[0]:          # first elements of both arrays are compared
            a.append(a.pop(0))       # If not equal, the first element of a is rotated to the end of the array
            time += 1
        a.pop(0)                   # here, the first elements of both arrays are equal, they are popped
        b.pop(0)
        time += 1
    return time

n = int(input().strip())
b = list(map(int, input().strip().split()))
a = list(map(int, input().strip().split()))
print(minimum_time(n, a, b))

```

```
3
1 3 2
2 3 1
6
```