Q1

class Solution:

def nextGreaterElement(self, nums1: List[int], nums2: List[int]) -> List[int]:

stack = []

nextGreaterDic = {ch:-1 for ch in nums2}

for i in range(len(nums2)):

while stack and nums2[stack[-1]] < nums2[i]:

nextGreaterDic[nums2[stack.pop()]] = nums2[i]

stack.append(i)

for i, ch in enumerate(nums1):

nums1[i] = nextGreaterDic[ch]

return nums1

Q2

def sorted(self, s):

# Code here

if not s:

return s

#take out the top element from stack

x = s.pop()

#sort the remaining stack

self.sorted(s)

tmpstack = []

while s and s[-1]>x:

tmpstack.append(s.pop())

s.append(x)

while tmpstack:

s.append(tmpstack.pop())

return s

Q3

class Solution:

def deleteMid(self, s, n):

if(n%2==1):

p=n//2

else:

p=(n//2)-1

st=[]

for i in range(p+1):

st.append(s.pop(0))

st.pop()

while(len(st)!=0):

s.insert(0,st.pop())

return s

Q5

st = [];

def push\_digits(number):

while (number != 0):

st.append(number % 10);

number = int(number / 10);

def reverse\_number(number):

# Function call to push number's

# digits to stack

push\_digits(number);

reverse = 0;

i = 1;

while (len(st) > 0):

reverse = reverse + (st[len(st) - 1] \* i);

st.pop();

i = i \* 10;

return reverse;

number = 39997;

print(reverse\_number(number));

Q6

from collections import deque

def reverse\_first\_k(q, k):

solve(q, k)

s = len(q) - k

for \_ in range(s):

x = q.popleft()

q.append(x)

return q

def solve(q, k):

if k == 0:

return

e = q.popleft()

solve(q, k - 1)

q.append(e)

queue = deque([10, 20, 30, 40, 50, 60, 70, 80, 90, 100])

k = 5

queue = reverse\_first\_k(queue, k)

while queue:

print(queue.popleft(), end=' ')

Q7

class Solution:

def removeAdj(self,v,n):

# Your code goes here

stack = []

for a in v:

if not len(stack): stack.append(a)

elif (stack[-1] == a): stack.pop()

else: stack.append(a)

return len(stack)

Q8

class Solution:

def findMaxDiff(self, arr, n):

left = [0]\*n

right = [0]\*n

l, r = [], []

r.append(0)

l.append(0)

for i in range(n-1,-1,-1):

while(arr[i]<=r[-1]):

r.pop()

right[i] = r[-1]

r.append(arr[i])

for i in range(0,n):

while(arr[i]<=l[-1]):

l.pop()

left[i] = l[-1]

l.append(arr[i])

ans = 0

for i in range(n):

val = abs(left[i] - right[i])

ans = max(ans,val)

return ans