Assignment - 4

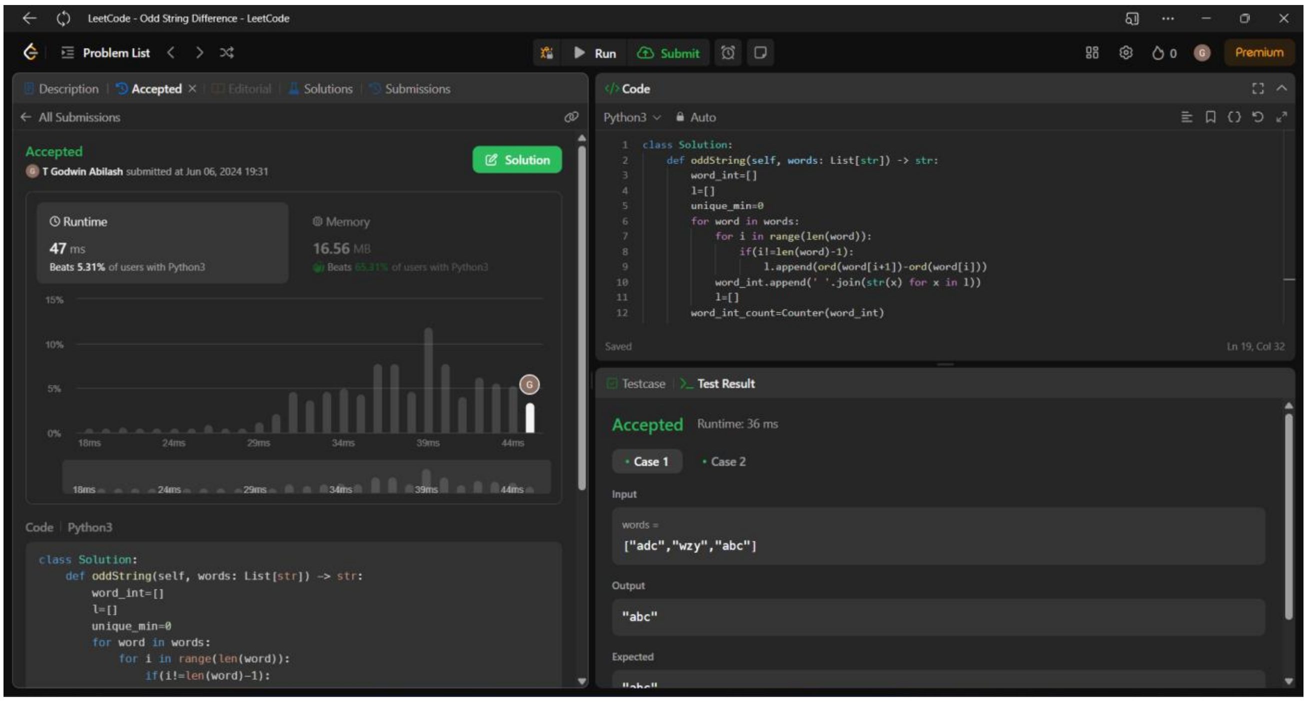
T Godwin Abilash – 192321012

1. Odd String Difference

Code:

```
class Solution:
    def oddString(self, words: List[str]) -> str:
        word_int=[]
        l=[]
        unique_min=0
        for word in words:
            for i in range(len(word)):
                if(i!=len(word)-1):
                    l.append(ord(word[i+1])-ord(word[i]))
            word_int.append(' '.join(str(x) for x in l))
            l=[]
        word_int_count=Counter(word_int)
        for i in word_int_count:
            if(word_int_count[i]==1):
                unique_min=i
                break
        for i in range(len(word_int)):
            if(word_int[i]==unique_min):
                return words[i]
```

Screenshot:



Accepted

T Godwin Abilash submitted at Jun 06, 2024 19:31

Runtime
47 ms
Beats 5.31% of users with Python3

Memory
16.56 MB
Beats 65.31% of users with Python3

```
class Solution:
    def oddString(self, words: List[str]) -> str:
        word_int=[]
        l=[]
        unique_min=0
        for word in words:
            for i in range(len(word)):
                if(i!=len(word)-1):
```

```
class Solution:
    def oddString(self, words: List[str]) -> str:
        word_int=[]
        l=[]
        unique_min=0
        for word in words:
            for i in range(len(word)):
                if(i!=len(word)-1):
                    l.append(ord(word[i+1])-ord(word[i]))
            word_int.append(' '.join(str(x) for x in l))
            l=[]
        word_int_count=Counter(word_int)
```

Accepted Runtime: 36 ms

Input

words =
["adc","wzy","abc"]

Output

"abc"

Expected

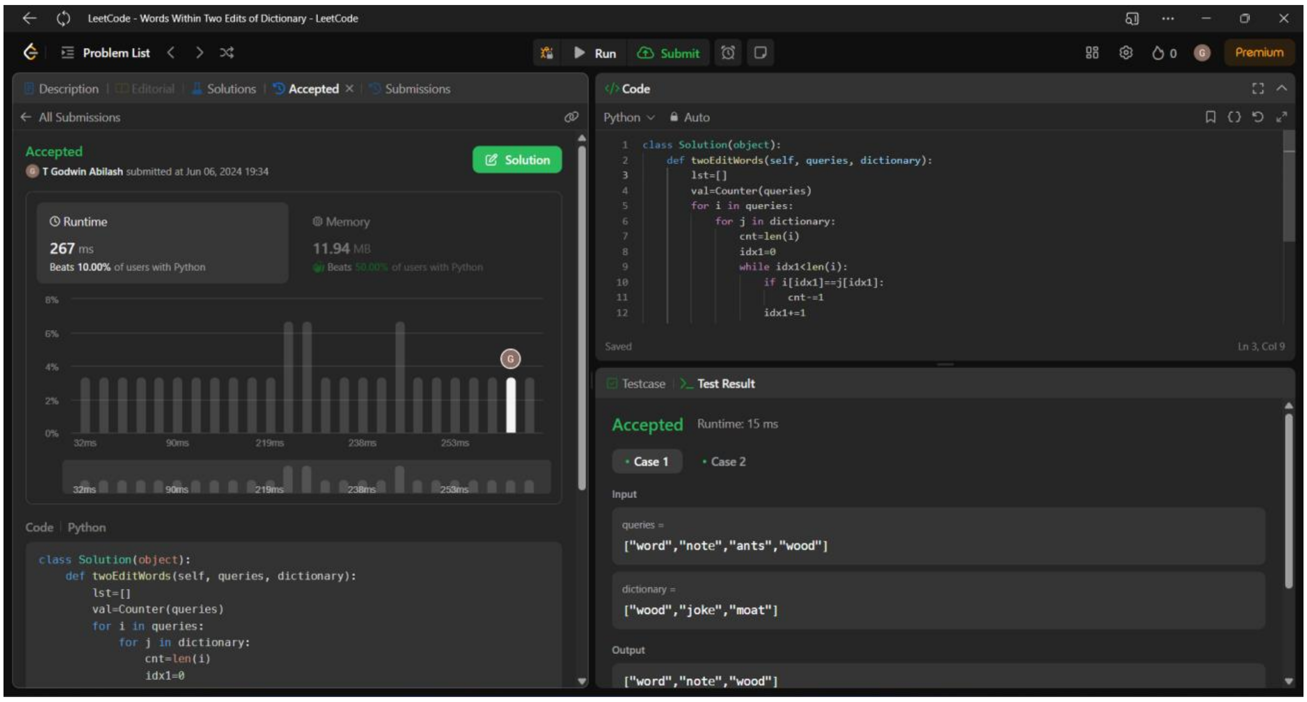
Time Complexity: O(n)

2. Words Within Two Edits of Dictionary

Code:

```python
class Solution(object):
    def twoEditWords(self, queries, dictionary):
        lst=[]
        val=Counter(queries)
        for i in queries:
            for j in dictionary:
                cnt=len(i)
                idx1=0
                while idx1<len(i):
                    if i[idx1]==j[idx1]:
                        cnt-=1
                    idx1+=1
                if cnt<=2 and val[i]!=0:
                    val[i]-=1
                    lst.append(i)
                    break
        return lst
```

Screenshot for I/O:


LeetCode - Words Within Two Edits of Dictionary - LeetCode

Accepted
T Godwin Abilash submitted at Jun 06, 2024 19:34

Runtime
267 ms
Beats 10.00% of users with Python

Memory
11.94 MB
Beats 50.00% of users with Python

Code | Python

```python
class Solution(object):
    def twoEditWords(self, queries, dictionary):
        lst=[]
        val=Counter(queries)
        for i in queries:
            for j in dictionary:
                cnt=len(i)
                idx1=0
```

```python
class Solution(object):
    def twoEditWords(self, queries, dictionary):
        lst=[]
        val=Counter(queries)
        for i in queries:
            for j in dictionary:
                cnt=len(i)
                idx1=0
                while idx1<len(i):
                    if i[idx1]==j[idx1]:
                        cnt-=1
                    idx1+=1
```

Saved   Ln 3, Col 9

Accepted Runtime: 15 ms

Case 1   Case 2

Input

queries =
["word","note","ants","wood"]

dictionary =
["wood","joke","moat"]

Output

["word","note","wood"]

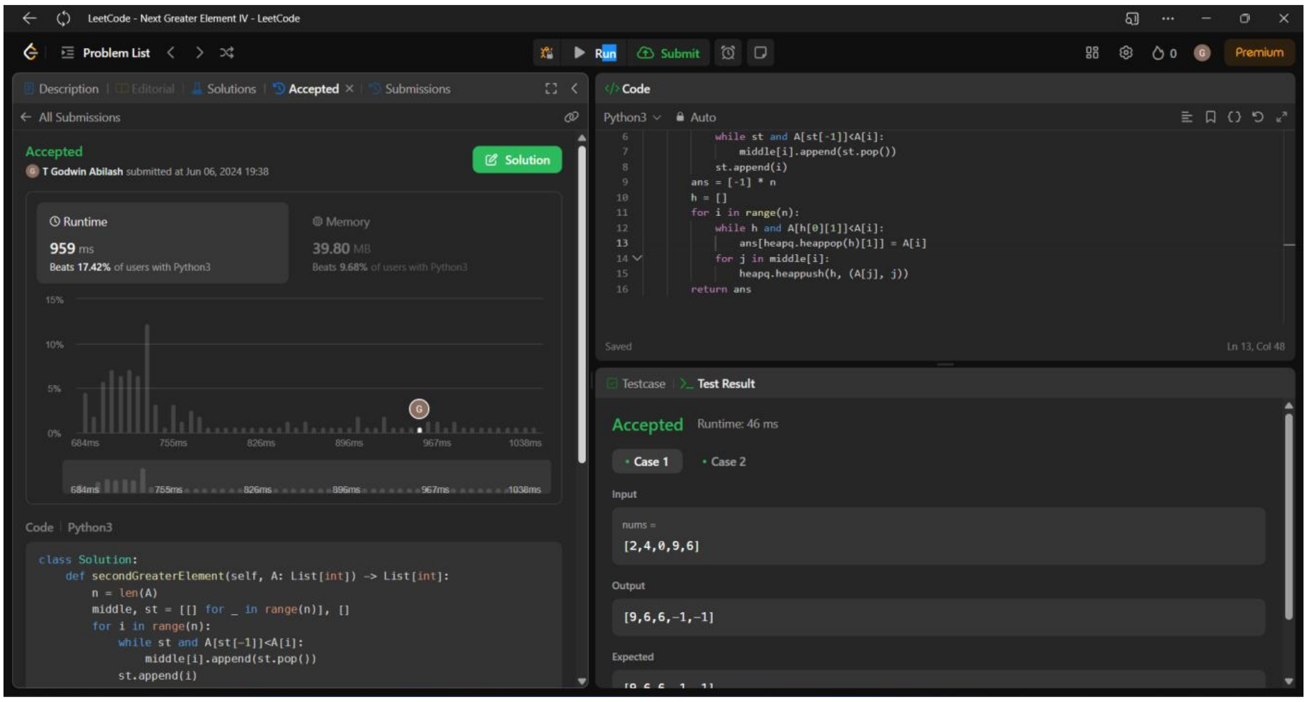
Time Complexity: O(n2)

3. Next Greater Element IV

Code:

```
class Solution:
    def secondGreaterElement(self, A: List[int]) -> List[int]:
        n = len(A)
        middle, st = [[] for _ in range(n)], []
        for i in range(n):
            while st and A[st[-1]]<A[i]:
                middle[i].append(st.pop())
            st.append(i)
        ans = [-1] * n
        h = []
        for i in range(n):
            while h and A[h[0][1]]<A[i]:
                ans[heapq.heappop(h)[1]] = A[i]
            for j in middle[i]:
                heapq.heappush(h, (A[j], j))
        return ans
```

Screenshot:



Accepted
T Godwin Abilash submitted at Jun 06, 2024 19:38

Runtime
959 ms
Beats 17.42% of users with Python3

Memory
39.80 MB
Beats 9.68% of users with Python3

Code | Python3

```
class Solution:
    def secondGreaterElement(self, A: List[int]) -> List[int]:
        n = len(A)
        middle, st = [[] for _ in range(n)], []
        for i in range(n):
            while st and A[st[-1]]<A[i]:
                middle[i].append(st.pop())
            st.append(i)
```

```
            while st and A[st[-1]]<A[i]:
                middle[i].append(st.pop())
            st.append(i)
        ans = [-1] * n
        h = []
        for i in range(n):
            while h and A[h[0][1]]<A[i]:
                ans[heapq.heappop(h)[1]] = A[i]
            for j in middle[i]:
                heapq.heappush(h, (A[j], j))
        return ans
```

Saved
Ln 13, Col 48

Accepted Runtime: 46 ms

Case 1 Case 2

Input

nums =
```
[2,4,0,9,6]
```

Output
```
[9,6,6,-1,-1]
```

Expected
```
[9,6,6,-1,-1]
```

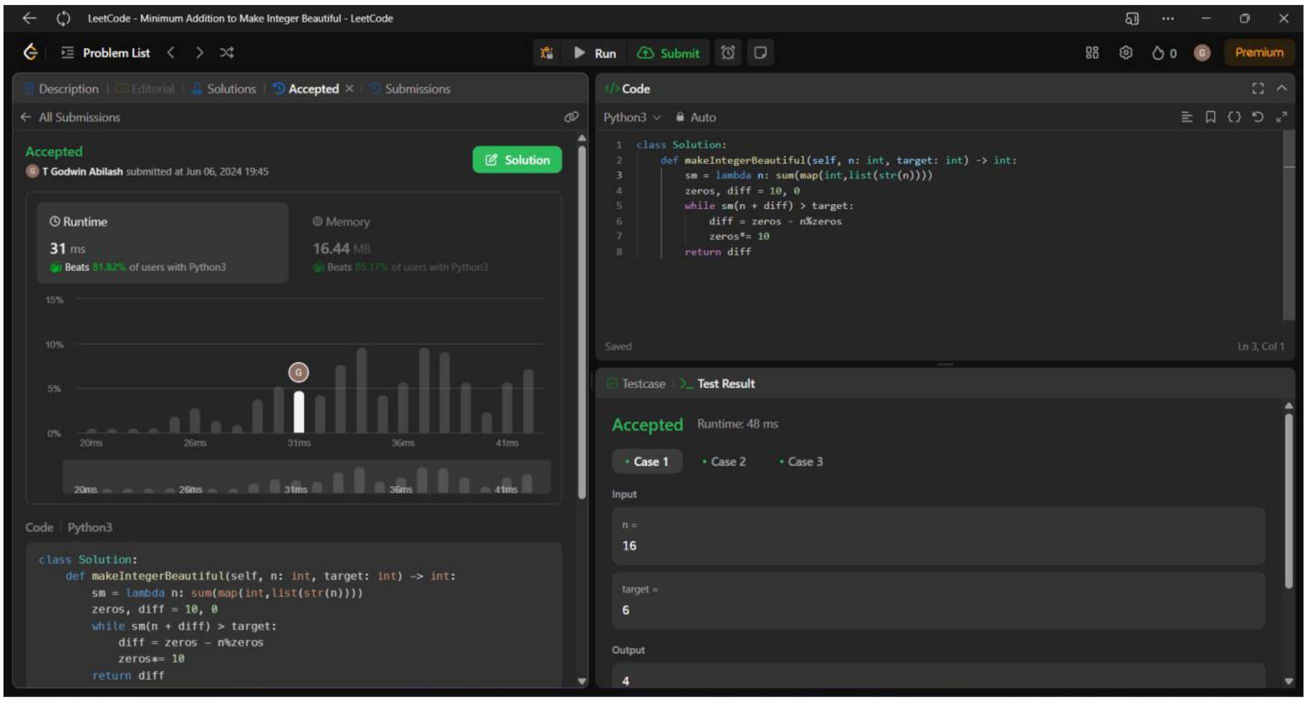
Time Complexity: O(n2)

4. Minimum Addition to make integer beautiful

Code:

```
class Solution:
    def makeIntegerBeautiful(self, n: int, target: int) -> int:
        sm = lambda n: sum(map(int,list(str(n))))
        zeros, diff = 10, 0
        while sm(n + diff) > target:
            diff = zeros - n%zeros
            zeros*= 10
        return diff
```

Screenshot:


LeetCode - Minimum Addition to Make Integer Beautiful - LeetCode

Accepted
T Godwin Abilash submitted at Jun 06, 2024 19:45

Runtime
31 ms
Beats 81.82% of users with Python3

Memory
16.44 MB
Beats 85.17% of users with Python3

Code | Python3

```
class Solution:
    def makeIntegerBeautiful(self, n: int, target: int) -> int:
        sm = lambda n: sum(map(int,list(str(n))))
        zeros, diff = 10, 0
        while sm(n + diff) > target:
            diff = zeros - n%zeros
            zeros*= 10
        return diff
```

Saved   Ln 3, Col 1

Accepted Runtime: 48 ms

Case 1 · Case 2 · Case 3

Input

n =
16

target =
6

Output

4

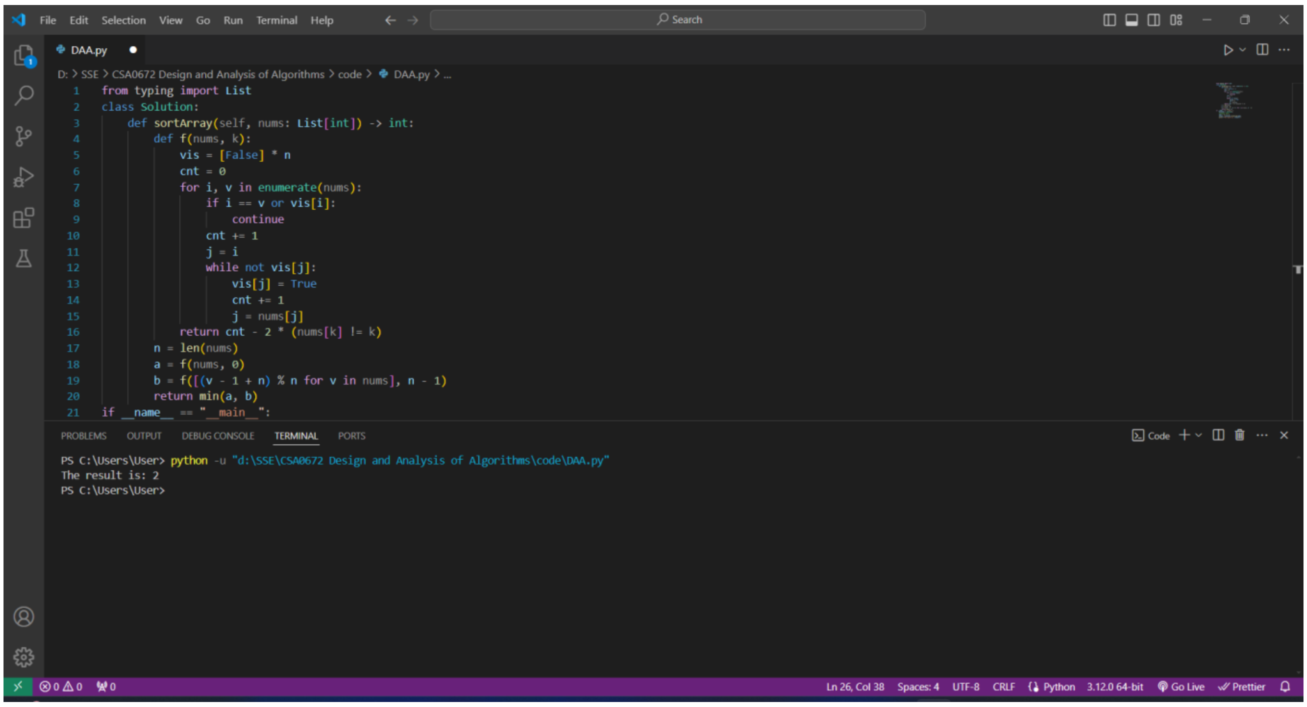
Time Complexity: O(n)

5. sort array by moving items to empty space

Code:

```
from typing import List
class Solution:
    def sortArray(self, nums: List[int]) -> int:
        def f(nums, k):
            vis = [False] * n
            cnt = 0
            for i, v in enumerate(nums):
                if i == v or vis[i]:
                    continue
                cnt += 1
                j = i
                while not vis[j]:
                    vis[j] = True
                    cnt += 1
                    j = nums[j]
            return cnt - 2 * (nums[k] != k)
        n = len(nums)
        a = f(nums, 0)
        b = f([(v - 1 + n) % n for v in nums], n - 1)
        return min(a, b)
if __name__ == "__main__":
    solution = Solution()
    # Example input
    nums = [2, 0, 1]
    result = solution.sortArray(nums)
    print(f"The result is: {result}")
```

Screenshot:



```
from typing import List
class Solution:
    def sortArray(self, nums: List[int]) -> int:
        def f(nums, k):
            vis = [False] * n
            cnt = 0
            for i, v in enumerate(nums):
                if i == v or vis[i]:
                    continue
                cnt += 1
                j = i
                while not vis[j]:
                    vis[j] = True
                    cnt += 1
                    j = nums[j]
            return cnt - 2 * (nums[k] != k)
        n = len(nums)
        a = f(nums, 0)
        b = f([(v - 1 + n) % n for v in nums], n - 1)
        return min(a, b)
if __name__ == "__main__":
```

```
PS C:\Users\User> python -u "d:\SSE\CSA0672 Design and Analysis of Algorithms\code\DAA.py"
The result is: 2
PS C:\Users\User>
```

Time Complexity: O(n2)