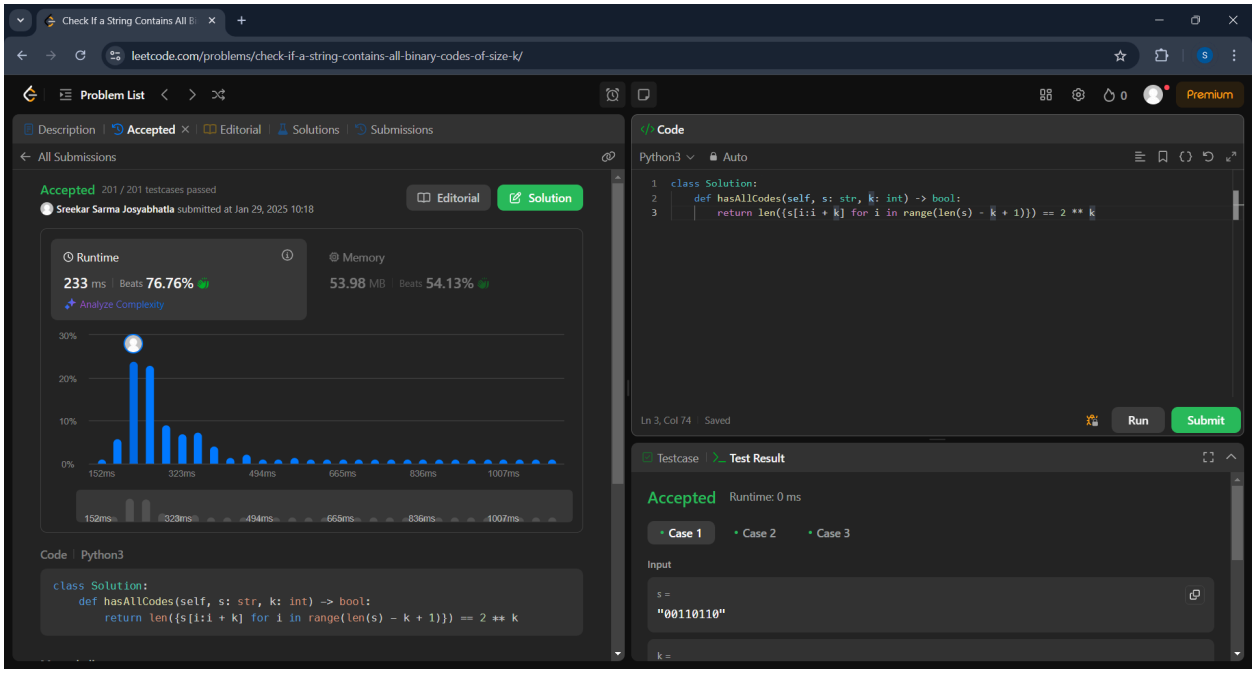
Assignment-3

J.Sreekar Sarma
VU21CSEN0300040

1. Check If a String Contains All Binary Codes of Size K

```python
class Solution:
    def hasAllCodes(self, s: str, k: int) -> bool:
        return len({s[i:i + k] for i in range(len(s) - k + 1)}) == 2 ** k
```

OUTPUT



2. Longest Chunked Palindrome Decomposition

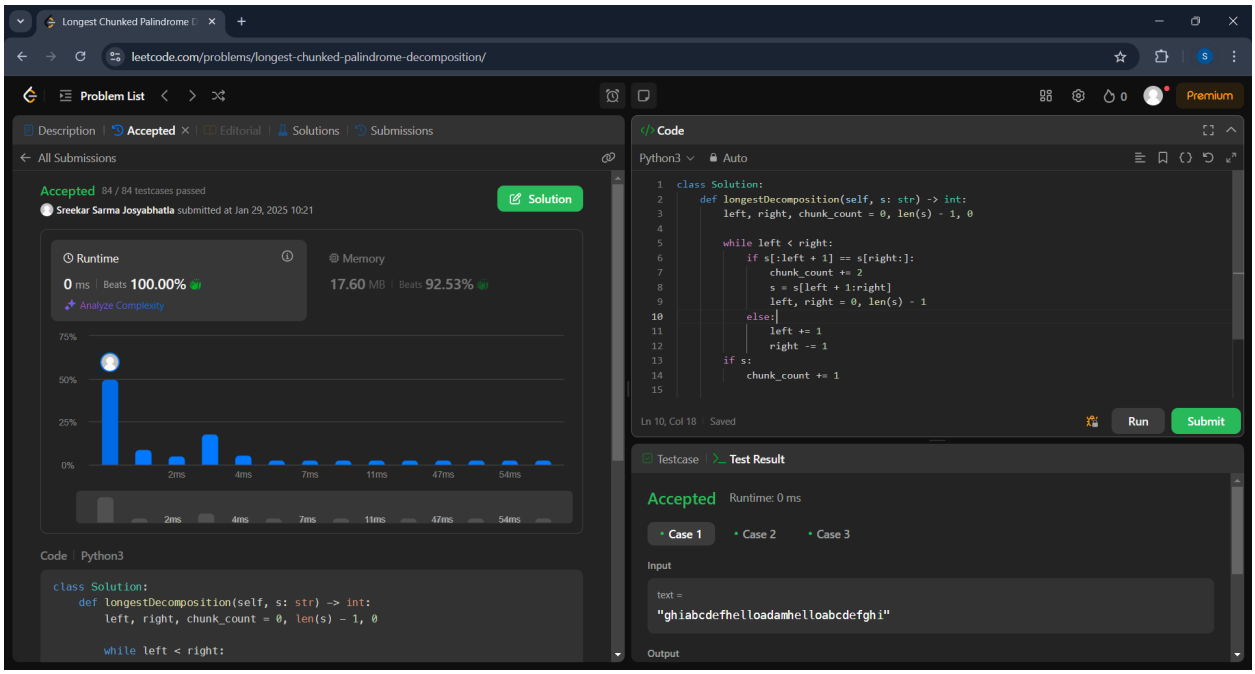
```python
class Solution:
    def longestDecomposition(self, s: str) -> int:
        left, right, chunk_count = 0, len(s) - 1, 0

        while left < right:
            if s[:left + 1] == s[right:]:
                chunk_count += 2
                s = s[left + 1:right]
                left, right = 0, len(s) - 1
            else:
                left += 1
                right -= 1
        if s:
```

```
            chunk_count += 1

        return chunk_count
```

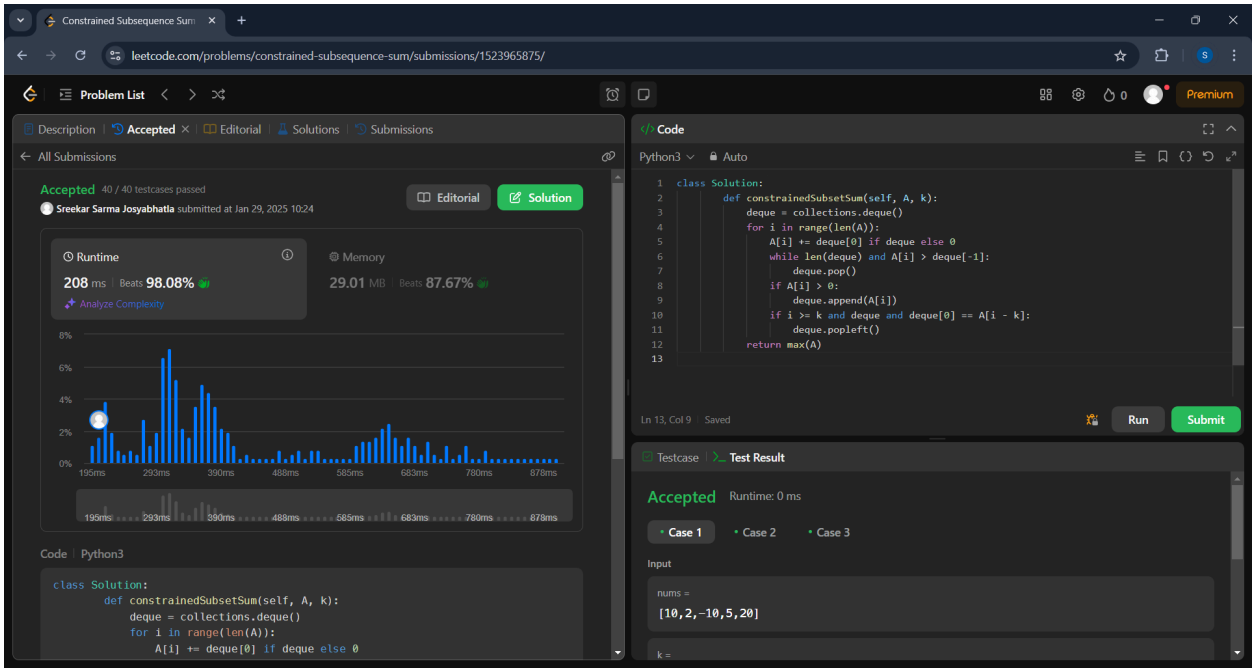
OUTPUT



3. Constrained Subsequence Sum

```
class Solution:
        def constrainedSubsetSum(self, A, k):
            deque = collections.deque()
            for i in range(len(A)):
                A[i] += deque[0] if deque else 0
                while len(deque) and A[i] > deque[-1]:
                    deque.pop()
                if A[i] > 0:
                    deque.append(A[i])
                if i >= k and deque and deque[0] == A[i - k]:
                    deque.popleft()
            return max(A)
```

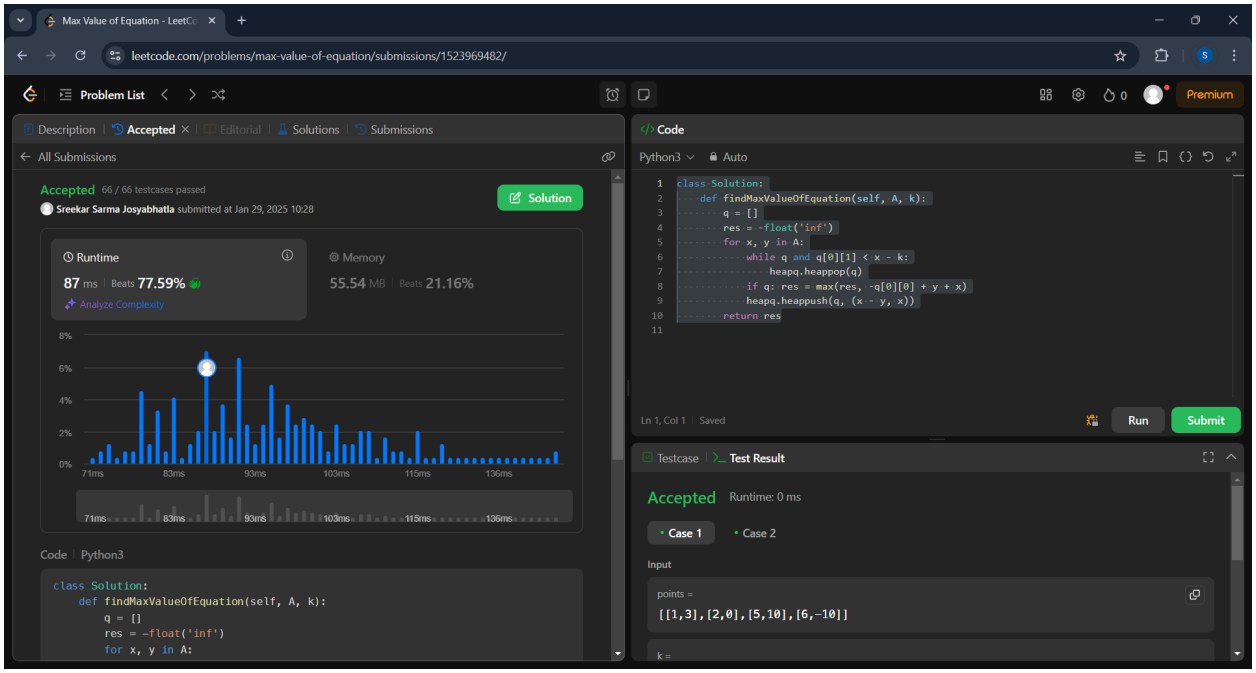
OUTPUT



4. Max Value of Equation

```python
class Solution:
    def findMaxValueOfEquation(self, A, k):
        q = []
        res = -float('inf')
        for x, y in A:
            while q and q[0][1] < x - k:
                heapq.heappop(q)
            if q: res = max(res, -q[0][0] + y + x)
            heapq.heappush(q, (x - y, x))
        return res
```

OUTPUT :


Max Value of Equation - LeetCo
leetcode.com/problems/max-value-of-equation/submissions/1523969482/
Problem List
Premium
Description | Accepted | Editorial | Solutions | Submissions
All Submissions
Accepted 66 / 66 testcases passed
Sreekar Sarma Josyabhatla submitted at Jan 29, 2025 10:28
Solution
Runtime
87 ms | Beats 77.59%
Analyze Complexity
Memory
55.54 MB | Beats 21.16%
8%
6%
4%
2%
0%
71ms
83ms
93ms
103ms
115ms
136ms
Code | Python3
class Solution:
def findMaxValueOfEquation(self, A, k):
q = []
res = -float('inf')
for x, y in A:
Code
Python3
Auto
class Solution:
def findMaxValueOfEquation(self, A, k):
q = []
res = -float('inf')
for x, y in A:
while q and q[0][1] < x - k:
heapq.heappop(q)
if q: res = max(res, -q[0][0] + y + x)
heapq.heappush(q, (x - y, x))
return res
Ln 1, Col 1
Saved
Run
Submit
Testcase | Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Input
points =
[[1,3],[2,0],[5,10],[6,-10]]
k =