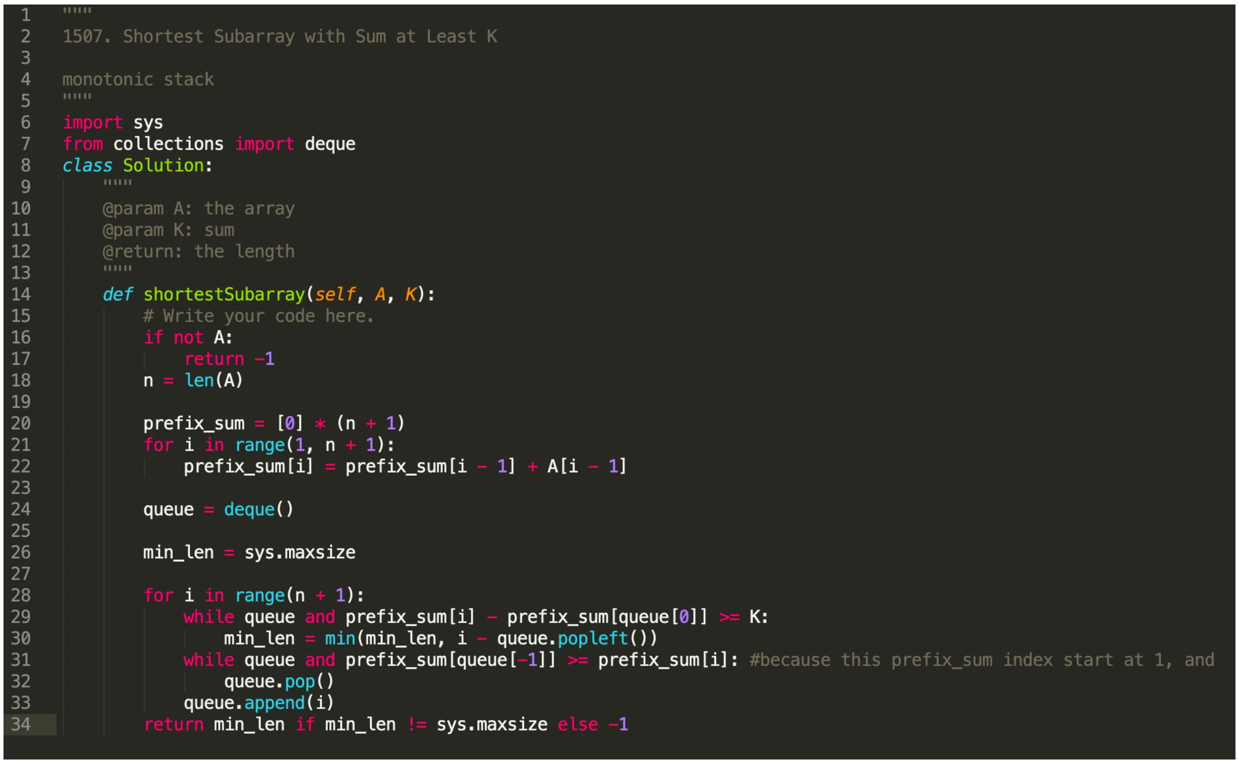
1507. Shortest Subarray with Sum at Least K (单调队列)

单调队列



```python
"""
1507. Shortest Subarray with Sum at Least K

monotonic stack
"""
import sys
from collections import deque
class Solution:
    """
    @param A: the array
    @param K: sum
    @return: the length
    """
    def shortestSubarray(self, A, K):
        # Write your code here.
        if not A:
            return -1
        n = len(A)

        prefix_sum = [0] * (n + 1)
        for i in range(1, n + 1):
            prefix_sum[i] = prefix_sum[i - 1] + A[i - 1]

        queue = deque()

        min_len = sys.maxsize

        for i in range(n + 1):
            while queue and prefix_sum[i] - prefix_sum[queue[0]] >= K:
                min_len = min(min_len, i - queue.popleft())
            while queue and prefix_sum[queue[-1]] >= prefix_sum[i]: #because this prefix_sum index start at 1, and
                queue.pop()
            queue.append(i)
        return min_len if min_len != sys.maxsize else -1
```

K=10

	✓		✓	
prefix_sum	3	7	13	14
index	0	1	2	3
num	3	4	6	1
	✗			

① 即使之后 prefix_sum 满足，则长度必然大于原来的 2−0 因此之前已经满足过的数是不需要存在 queue 中的。

	✓		✓			
prefix_sum	3	7	13	14	9	19
index	0	1	2	3	4	?
num	3	4	6	1	-5	
			✗	✗		

② 当前加入 -5，则 prefix_sum =9。即使后面满足，如出现 prefix_sum=19，则必然会选则 9 而不会选则更远的 13, 14。因此当前插入的 prefix_sum 比 queue 尾的要小，则 pop