

Sliding Window: Dynamic

Problem: Given an array of positive integers and a target sum, find a contiguous subarray whose sum is as close as possible to, but does not exceed, the target.

Example: nums = [1, 3, 2, 5, 1, 1, 2, 1, 4], target = 8



Window: [0, -1] Sum: 0 Target: 8
Start dynamic sliding window

```
def find_all_subarrays(nums, target):  
    """  
    Finds and prints all contiguous subarrays whose sum equals the target.  
    Assumes that all numbers in 'nums' are positive.  
    """  
  
    print("Problem: Given an array of positive integers and a target sum,")  
    print("find all contiguous subarrays whose sum equals the target.\n")  
    print(f"Example: nums = {nums}, target = {target}\n")  
    print("-" * 70)  
  
    left = 0  
    current_sum = 0  
    n = len(nums)  
  
    # Iterate with 'right' pointer to expand the window.  
    for right in range(n):  
        current_sum += nums[right]  
        print(f"Expanding window: added nums[{right}] = {nums[right]}")  
        print(f"Current window (indices [{left} to {right}]): {nums[left:right+1]}")  
        print(f"Current sum: {current_sum}\n")  
  
        # If the sum exceeds the target, contract the window from the left.
```

```
while current_sum > target and left <= right:
    print(f" Sum {current_sum} exceeds target {target}.")
    print(f" Removing nums[{left}] = {nums[left]} from window.")
    current_sum -= nums[left]
    left += 1
    print(f" After contraction, window (indices [{left} to {right}]): {nums[left:right+1]}")
    print(f" Current sum: {current_sum}\n")

# If the current sum equals the target, print the subarray.
if current_sum == target:
    print(f"Found subarray with sum {target}: indices [{left}, {right}] -> {nums[left:right+1]}\n")

print("-" * 70)
print("Done searching for subarrays.")

# Example usage:
nums = [1, 3, 2, 5, 1, 1, 2, 1, 4]
target = 8
find_all_subarrays(nums, target)
```

Revision #3

Created 28 December 2024 22:15:37 by victor

Updated 16 March 2025 23:14:29 by victor