**Problem Name: Two Sum III design**

**Topics:** Hash Table, Linked list, Design, Doubly linked list

**Companies:** Amazon, Microsoft, Google, Linkedin, Salesforce, Snapchat, Facebook, Bloomberg, Walmart, Arcesium.

**Level: Hard**

**Language:** C++

**Problem Statement:**

**Input Format:**

First line of input contains integer value capacity

Second line of input contains integer value n (total no of “put” + “get” operation)

Next n line contains:

1st line is string “put” or “get” representing function to call

If it’s “put” then next line contains two integers key and value

If it’s “get” then next line contains integer value key

Ex:

2

10

put

1 1

put

2 2

get

1

put

3 3

get

2

get

3

put

4 4

get

1

get

3

get

4

**Output Format:** Print output from function get whenever it is called. Ex for above input output would be

1 -1 3 -1 3 4d

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:** Two pointer method, to beginning and end of the vector by sorting.

**Code:**

**Time Complexity**: O(nlogn)

**Space Complexity:** O(n)

**Optimized Solution:**

* **Explanation:** in this approach, it is used an unordered\_map to lookup if the difference of the target sum and previous found number is present. The reason for a map (rather than a set), is to handle case where the sum would be the double of a present number.
* For better performance, we exploit the given number range from the problem specification, to avoid, when applicable, the cost of the enumerating through the map and do a lookup for each.

**Code:**

**Time Complexity**: O(n) even though hashmap works in O(1) time complexity but we are calling it n times.

**Space Complexity:** O(n)