1. **2 Sum**

**Link:** <https://leetcode.com/problems/two-sum/>

**Approach 1:** Brute force – O(n^2) i.e., check every pair

**Approach 2:** Sort the array O(nLogn), for each element find the required value using binary search(avoid taking same number twice) for n elements(nLogn) hence total O(nLogn). **OR**

Use two pointer approach. O(nLogN + n): sort and check

If(s+e==t) success s++,e—

If(s+e<t) s++;

If(s+e>t) e--;

**Approach 3:** Use unordered set.

1. Pick a value from array check if required value is present in set.
2. If not present insert this value also.

If we have to return indices sorting will not work.

1. **3 Sum**

**Link:** <https://leetcode.com/problems/3sum/>

Divide in two parts: Value + 2 sum(Using two pointer);

For unique store triplets in set **OR**

Keep incrementing s , i and decrementing e till they are same.