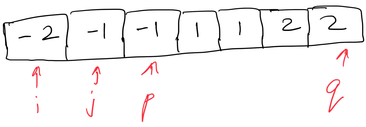
4 Sum

arr = [-2, -1, -1, 1, 1, 2, 2]

won't always be sorted

tar = 0

i ≠ j ≠ p ≠ q return i, j, p, q



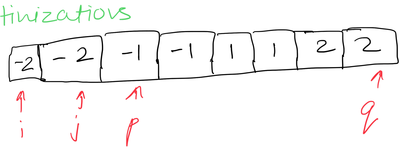
if i+j+p+q < target
p++

i+j+p+q > target
q--

i+j+p+q = target
return ans
p++, q--

optimizations

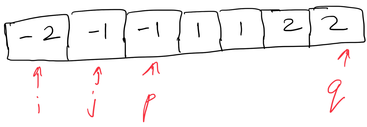
1



i=0 & value = -2
and if we find answer we do
i++ so i=1 & value = -2 again
we should skip this index because
we already have an answer with i=-2
and we need to avoid duplicate answers

```
if(i>0 && nums[i]==nums[i-1])
    continue;
```

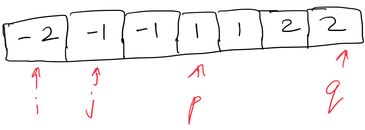
2



j = -1 for this iteration and
if we find out that j = -1 in the
next iteration also so skip it otherwise
we will get duplicate answer

```
while(j<n && nums[j]==nums[j-1])
    j++;
```

3



p=1 this time next time also p=1 so
skip

```
while(p<q && nums[p]==nums[q-1])
    p++;
```

Code

```
sort
for (i=0 to n){
    if(i>0 && nums[i]==nums[i-1])
        continue;
    for(j=i+1 to n) {
        p = j+1
        q = n-1
        while (p<q) {
            sum = i+j+p+q
            if(sum <tar) p++
            else if (sum>tar) q--
            else {
                return ans
                p++, q--
                while(p<q && nums[p]==nums[q-1]){
                    p++;
                }
            }
        }
        j++
        while(j<n && nums[j]==nums[j-1]) {
            j++;
        }
    }
}
return ans
```

TC: O(nlogn + n^3)