Question: https://leetcode.com/problems/4sum/

The problem is similar to 3 sum problem, in 3 sum we use a loop to first take 1st element as pivot and here in 4 sum we use 2 nested loop to consider first 2 elements as pivot and search for the last elements like 2 sum.

Code:  
class Solution {

public List<List<Integer>> fourSum(int[] nums, int target) {

Arrays.sort(nums);

List<List<Integer>> res = new ArrayList<>();

for(int i=0; i<nums.length; i++){

for(int j=i+1; j<nums.length; j++){

int s=j+1, e=nums.length-1, k=target-nums[i]-nums[j];

while(s<e){

int sum = nums[s]+nums[e];

if(k==sum){

List<Integer> sub=new ArrayList<Integer>();

sub.add(nums[i]);

sub.add(nums[j]);

sub.add(nums[s]);

sub.add(nums[e]);

res.add(sub);

s++;

e--;

while(s<e && nums[s]==sub.get(2))s++;

while(s<e && nums[e]==sub.get(3))e--;

}

else if(k>sum) s++;

else if(k<sum) e--;

}

while(j<nums.length-1 && nums[j]==nums[j+1]) j++;

}

while(i<nums.length-1 && nums[i]==nums[i+1]) i++;

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>