**Problem Name:** Four sum

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

First line of the input contain integer n (size of list)

Second line contain n space separated integer list values.

Last line contain integer value pos representing value of node to delete.

Ex:

5

1 2 3 4 5

1

**Output Format:** Print linked list after removing node having value pos

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:**

* First, we’ll sort the array.
* Take 4 pointers: ***i, j, left, right.***
* Outer 2 loops for ***i & j.*** We store the remaining value to find in sum variable.
* Then we try to calculate the ***left+right*** values & if they are equal then push all 4 values to the set.
* If the value is less than sum then we’ll increase left because array is in sorted order, else we’ll decrease right.

**Code:**

**Time Complexity**: O(n^3 logn).

**Space Complexity:**

**Optimized Solution:**

**Explanation:**

1. First we store in a map twoSums all the pairs of numbers, while the key is their sum and the value is a pair of indices.
2. For each twoSum, we check if we have its complementary to target in the map too.
3. If so, we take the lists of indices of the key and the target-key, and pass them to the function makePairs.
4. In makePairs we loop through the pairs of indices of both lists and check if the indices are not the same. That's because we don't want to use the same index twice.
5. We sort the four results to avoid duplicate fourSums in different orders.
6. Insert into a set to avoid duplicate fourSums.
7. Last step is to convert the set into a vector and here we got our result

**Code:**

**Time Complexity**: O(n^2 logn)  time complexity of set insertion is O(log n), so the total time complexity is expected to be O(n^2 log n).

**Space Complexity:** O(n)