

# 4Sum

Given an array `nums` of `n` integers and an integer `target`, are there elements `a`, `b`, `c`, and `d` in `nums` such that `a + b + c + d = target`? Find all unique quadruplets in the array which gives the sum of `target`.

Note:

The solution set must not contain duplicate quadruplets.

Example:

Given array `nums = [1, 0, -1, 0, -2, 2]`, and `target = 0`.

A solution set is:

```
[
    [-1, 0, 0, 1],
    [-2, -1, 1, 2],
    [-2, 0, 0, 2]
]
```

---

Solution in C++

```
class Solution {
public:
    vector<vector<int>> fourSum(vector<int>& nums, int target) {
        sort(nums.begin(), nums.end());

        map<int,int> mp;
        for(int i=0; i<nums.size(); i++) mp[nums[i]]=i+1;

        set<vector<int>> sRet; // storage will contain results

        for(int i=0; i<nums.size(); i++)
            for(int j=i+1; j<nums.size(); j++)
                for(int k=j+1; k<nums.size(); k++) {
                    int l = target - nums[i] - nums[j] - nums[k];
                    if (mp[l] && (mp[l]-1)>k) {
                        vector<int> tmp = {nums[i], nums[j], nums[k], l};
                        sRet.insert(tmp);
                    }
                }
        vector<vector<int>> ret;
        for(auto n:sRet) ret.push_back(n);
        return ret;
    }
};
```