Two Sum

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        int i;
        int j;

        vector<int> awn;

        for (i = 0;i<nums.size();i++){
            if (nums[i] <= target){
              for (j = i+1;j<nums.size();j++){
                if ((nums[j] + nums[i]) == target){
                    awn.push_back(i);
                    awn.push_back(j);
                    return awn;
                }
              }
            }
        }

        for (i = 0;i<nums.size();i++){
            if (nums[i] >= target){
              for (j = i+1;j<nums.size();j++){
                if ((nums[j] + nums[i]) == target){
                    awn.push_back(i);
                    awn.push_back(j);
                    return awn;
                }
              }
            }
        }

        return awn;
    }
};
```

- Working solution
    - iterates through and finds numbers less then target
    - then compares with numbers after to see if they add up to target
    - if nothing is found
    - iterates through and finds numbers greater then target (negative target)
    - then compares with numbers after to see if they add up to target
- Big O
    - O(n^2)

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        int i;
        vector<int> awn;
        unordered_map<int,int> map;
        int need;

        for (i = 0;i<nums.size();i++){
            need = target - nums[i];
            if (map.count(need)){
              awn.push_back(map[need]);
              awn.push_back(i);
              return awn;
            }
            else{
              map[nums[i]] = i;
            }
        }

        return awn;
    }
};
```

- Improved solution

- first solution works how ever it scored 111 ms which is slow
- this solution scored 8ms
- takes current index's element and calculates what it needs to reach the target
- checks the hash map if the needed number is in it
- if not, it stores the number/element in the hash table with the index as the value