Two Sum (1)

Use hashmap to store the value in vector as key and it's index as the value in the map. Then set val = target - nums[i]. If val exists in hashmap, return the indices.

Time complexity → O(n).
space complexity → O(n).

Code

```
vector <int> twoSum (vector <int> nums, int target) {
        vector <int> v;
        unordered_map < int, int > umap;

        for ( int i=0 ; i< nums.size(); i++) {
                umap [nums[i]] = i;
        }

        for ( int i=0 ; i < nums.size() ; i++ ) {
                int val = target - nums[i];
                if (umap.find(val) != umap.end() &&
                                i != umap[val] ) {

                        v.push_back (i);
                        v.push_back (umap[val]);
                        return v;
                }
        }
        return v;    }.
```