

Contents

1 数组	1
1.1 两数之和	1
2 链表	2
3 树	2
4 图	2
5 动态规划	2

1 数组

1.1 两数之和

题目:

给定一个整数数组和一个目标值, 找出数组中和为目标值的两个数。

你可以假设每个输入只对应一种答案, 且同样的元素不能被重复利用。

示例:

给定 `nums = [2, 7, 11, 15]`, `target = 9`

因为 `nums[0] + nums[1] = 2 + 7 = 9` 所以返回 `[0, 1]`

解法:

用一个 `map`, `key` 是元素值, `value` 是 `idx` 看新来的这个元素的目标值 (`tgt - nums[i]`) 在不在 `map` 里, 在的话把它的 `value` 拿出来就行了。。

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        vector<int> res;
        unordered_map<int, int> map;
        for (int i = 0; i < nums.size(); ++i) {
            const int& tgt_val = target - nums[i];
            if (map.find(tgt_val) != map.end()) {
                res.push_back(map[tgt_val]);
                res.push_back(i);
                return res;
            } else {
                map.insert(std::make_pair(nums[i], i));
            }
        }
    }
};
```

- 2 链表
- 3 树
- 4 图
- 5 动态规划