
LeetCodeCrowdsource

发布 1.0

LeetCodeCrowdsource

2020 年 03 月 08 日

1	题解	1
1.1	[0001] 两数之和	1
1.2	[0002] 两数相加	3
1.3	[0003] 无重复字符的最长子串	4
1.4	[0004] 寻找两个有序数组的中位数	6
1.5	[0005] 最长回文子串	7
1.6	[0006] Z 字形变换	9
1.7	[0007] 整数反转	10
1.8	[0008] 字符串转换整数 (atoi)	12
1.9	[0009] 回文数	13
1.10	[0010] 正则表达式匹配	15
1.11	[0011] 盛最多水的容器	16
1.12	[0012] 整数转罗马数字	18
1.13	[0013] 罗马数字转整数	19
1.14	[0014] 最长公共前缀	21
1.15	[0015] 三数之和	22
1.16	[0016] 最接近的三数之和	24
1.17	[0017] 电话号码的字母组合	25
1.18	[0018] 四数之和	27
1.19	[0019] 删除链表的倒数第 N 个节点	28
1.20	[0020] 有效的括号	30
2	专题	33
2.1	专题-BFS 算法	33
2.2	专题-DFS 算法	33
2.3	专题-DP 算法	33
2.4	专题-STL 算法	34
2.5	专题-二分算法	34

1.1 [0001] 两数之和

- <https://leetcode-cn.com/problems/two-sum>

1.1.1 题目描述

1.1.2 题目代码

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

1.1.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.2 [0002] 两数相加

- <https://leetcode-cn.com/problems/add-two-numbers>

1.2.1 题目描述

1.2.2 题目代码

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
    }
};
```

1.2.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.3 [0003] 无重复字符的最长子串

- <https://leetcode-cn.com/problems/longest-substring-without-repeating-characters>

1.3.1 题目描述

1.3.2 题目代码

```
class Solution {  
public:  
    int lengthOfLongestSubstring(string s) {  
    }  
};
```


1.3.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.4 [0004] 寻找两个有序数组的中位数

- <https://leetcode-cn.com/problems/median-of-two-sorted-arrays>

1.4.1 题目描述

1.4.2 题目代码

```
class Solution {  
public:  
    double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {  
    }  
};
```

1.4.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.5 [0005] 最长回文子串

- <https://leetcode-cn.com/problems/longest-palindromic-substring>

1.5.1 题目描述

1.5.2 题目代码

```
class Solution {  
public:  
    string longestPalindrome(string s) {  
    }  
};
```

1.5.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.6 [0006] Z 字形变换

- <https://leetcode-cn.com/problems/zigzag-conversion>

1.6.1 题目描述

1.6.2 题目代码

```
class Solution {  
public:  
    string convert(string s, int numRows) {  
  
    }  
};
```

1.6.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.7 [0007] 整数反转

- <https://leetcode-cn.com/problems/reverse-integer>

1.7.1 题目描述

1.7.2 题目代码

```
class Solution {  
public:  
    int reverse(int x) {  
    }  
};
```

1.7.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.8 [0008] 字符串转换整数 (atoi)

- atoi

1.8.1 题目描述

1.8.2 题目代码

```
class Solution {  
public:  
    int myAtoi(string str) {  
  
    }  
};
```

1.8.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```


方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.9 [0009] 回文数

- <https://leetcode-cn.com/problems/palindrome-number>

1.9.1 题目描述

1.9.2 题目代码

```
class Solution {  
public:  
    bool isPalindrome(int x) {  
    }  
};
```

1.9.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.10 [0010] 正则表达式匹配

- <https://leetcode-cn.com/problems/regular-expression-matching>

1.10.1 题目描述

1.10.2 题目代码

```
class Solution {  
public:  
    bool isMatch(string s, string p) {  
    }  
};
```

1.10.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.11 [0011] 盛最多水的容器

- <https://leetcode-cn.com/problems/container-with-most-water>

1.11.1 题目描述

1.11.2 题目代码

```
class Solution {  
public:  
    int maxArea(vector<int>& height) {  
    }  
};
```

1.11.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.12 [0012] 整数转罗马数字

- <https://leetcode-cn.com/problems/integer-to-roman>

1.12.1 题目描述

1.12.2 题目代码

```
class Solution {  
public:  
    string intToRoman(int num) {  
  
    }  
};
```

1.12.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.13 [0013] 罗马数字转整数

- <https://leetcode-cn.com/problems/roman-to-integer>

1.13.1 题目描述

1.13.2 题目代码

```
class Solution {  
public:  
    int romanToInt(string s) {  
    }  
};
```

1.13.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```


1.14 [0014] 最长公共前缀

- <https://leetcode-cn.com/problems/longest-common-prefix>

1.14.1 题目描述

1.14.2 题目代码

```
class Solution {  
public:  
    string longestCommonPrefix(vector<string>& strs) {  
    }  
};
```

1.14.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.15 [0015] 三数之和

- <https://leetcode-cn.com/problems/3sum>

1.15.1 题目描述

1.15.2 题目代码

```
class Solution {  
public:  
    vector<vector<int>> threeSum(vector<int>& nums) {  
    }  
};
```

1.15.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.16 [0016] 最接近的三数之和

- <https://leetcode-cn.com/problems/3sum-closest>

1.16.1 题目描述

1.16.2 题目代码

```
class Solution {  
public:  
    int threeSumClosest(vector<int>& nums, int target) {  
    }  
};
```

1.16.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.17 [0017] 电话号码的字母组合

- <https://leetcode-cn.com/problems/letter-combinations-of-a-phone-number>

1.17.1 题目描述

1.17.2 题目代码

```
class Solution {  
public:  
    vector<string> letterCombinations(string digits) {  
    }  
};
```

1.17.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.18 [0018] 四数之和

- <https://leetcode-cn.com/problems/4sum>

1.18.1 题目描述

1.18.2 题目代码

```
class Solution {  
public:  
    vector<vector<int>> fourSum(vector<int>& nums, int target) {  
    }  
};
```

1.18.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.19 [0019] 删除链表的倒数第 N 个节点

- <https://leetcode-cn.com/problems/remove-nth-node-from-end-of-list>

1.19.1 题目描述

1.19.2 题目代码

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* removeNthFromEnd(ListNode* head, int n) {

    }
};
```


1.19.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

1.20 [0020] 有效的括号

- <https://leetcode-cn.com/problems/valid-parentheses>

1.20.1 题目描述

1.20.2 题目代码

```
class Solution {  
public:  
    bool isValid(string s) {  
  
    }  
};
```

1.20.3 题目解析

方法一

分析

思路

注意

知识点

复杂度

参考

答案

```
//
```

方法二

分析

思路

注意

知识点

复杂度

参考

答案

//

2.1 专题-BFS 算法

2.1.1 引言

2.1.2 例子

2.2 专题-DFS 算法

2.2.1 引言

2.2.2 例子

2.3 专题-DP 算法

2.3.1 引言

2.3.2 例子

2.4 专题-STL 算法

2.4.1 引言

2.4.2 例子

2.5 专题-二分算法

2.5.1 引言

2.5.2 例子