

LintCode 参考程序

Lionx

2016 年 4 月 28 日

目录

第一章 入门 (Naive)	5
1.1 Problem ID: 228 Middle of Linked List	6
1.1.1 Description	6
1.1.2 Example	6
1.1.3 Code	6
1.2 Problem ID: 366 Fibonacci	8
1.2.1 Description	8
1.2.2 Example	8
1.2.3 Code	8
1.3 Problem ID: 452 Remove Linked List Elements	10
1.3.1 Description	10
1.3.2 Example	10
1.3.3 Code	10
第二章 容易 (Easy)	13
2.1 Problem ID: 1 A + B Problem	14
2.1.1 Description	14
2.1.2 Clarification	14
2.1.3 Example	14
2.1.4 Code	14
2.2 Problem ID: 2 Trailing Zeros	15
2.2.1 Description	15
2.2.2 Example	15
2.2.3 Code	15

2.3	Problem ID: 6 Merge Sorted Array II	16
2.3.1	Description	16
2.3.2	Example	16
2.3.3	Code	16
2.4	Problem ID: 8 Rotate String	17
2.4.1	Description	17
2.4.2	Example	17
2.4.3	Code	18
2.5	Problem ID: 9 Fizz Buzz	18
2.5.1	Description	18
2.5.2	Example	19
2.5.3	Code	19
2.6	Problem ID: 13 strStr	20
2.6.1	Description	20
2.6.2	Clarification	20
2.6.3	Example	20
2.6.4	Code	21
2.7	Problem ID: 14 First Position of Target	22
2.7.1	Description	22
2.7.2	Example	22
2.7.3	Code	22
2.8	Problem ID: 28 Search a 2D Matrix	23
2.8.1	Description	23
2.8.2	Example	23
2.8.3	Code	24
2.9	Problem ID: 82 Single Number	25
2.9.1	Description	25
2.9.2	Example	25
2.9.3	Code	26
2.10	Problem ID: 157 Unique Characters	26
2.10.1	Description	26
2.10.2	Example	26
2.10.3	Code	27

2.11	Problem ID: 166 Nth to Last Node in List	27
2.11.1	Description	27
2.11.2	Example	28
2.11.3	Code	28
2.12	Problem ID: 204 Singleton	29
2.12.1	Description	29
2.12.2	Example	30
2.12.3	Code	30
2.13	Problem ID: 496 Toy Factory	31
2.13.1	Description	31
2.13.2	Example	31
2.13.3	Code	31
2.14	Problem ID: 497 Shape Factory	33
2.14.1	Description	33
2.14.2	Example	33
2.14.3	Code	33
第三章	中等 (Medium)	37
3.1	Problem ID: 3 Digit Counts	38
3.1.1	Description	38
3.1.2	Example	38
3.1.3	Code	38
3.2	Problem ID: 4 Ugly Number II	40
3.2.1	Description	40
3.2.2	Example	40
3.2.3	Code	40
3.3	Problem ID: 5 Kth Largest Element	42
3.3.1	Description	42
3.3.2	Example	42
3.3.3	Code	42
3.4	Problem ID: 83 Single Number II	43
3.4.1	Description	43
3.4.2	Example	43
3.4.3	Code	43

3.5	Problem ID: 84 Single Number III	45
3.5.1	Description	45
3.5.2	Example	45
3.5.3	Code	45
3.6	Problem ID: 162 Set Matrix Zeroes	47
3.6.1	Description	47
3.6.2	Example	47
3.6.3	Code	47
3.7	Problem ID: 394 Coins in a Line	49
3.7.1	Description	49
3.7.2	Example	49
3.7.3	Code	49
3.8	Problem ID: 419 Roman to Integer	51
3.8.1	Description	51
3.8.2	Clarification	51
3.8.3	Example	51
3.8.4	Code	51
3.9	Problem ID: 428 Pow(x, n)	53
3.9.1	Description	53
3.9.2	Example	53
3.9.3	Code	53
第四章	困难 (Hard)	55
第五章	超难 (Super)	57

第一章 入门 (Naive)

1.1 Problem ID: 228 Middle of Linked List

1.1.1 Description

Find the middle node of a linked list.

1.1.2 Example

Given 1->2->3, return the node with value 2.

Given 1->2, return the node with value 1.

1.1.3 Code

C++

```
1  /**
2   * Definition of ListNode
3   * class ListNode {
4   * public:
5   *     int val;
6   *     ListNode *next;
7   *     ListNode(int val) {
8   *         this->val = val;
9   *         this->next = NULL;
10  *     }
11  * }
12  */
13  class Solution{
14  public:
15      /**
16       * @param head: the head of linked list.
17       * @return: a middle node of the linked list
18       */
19      ListNode *middleNode(ListNode *head) {
20          // Write your code here
21          if(head == NULL){
22              return NULL;
23          }
24          ListNode *fast = head;
25          ListNode *slow = head;
26          while(fast->next != NULL && fast->next->next != NULL){
27              slow = slow->next;
28              fast = fast->next->next;
29          }
30          return slow;
31      }
32  };
```


Python

```
1  """
2  Definition of ListNode
3  class ListNode(object):
4
5      def __init__(self, val, next=None):
6          self.val = val
7          self.next = next
8  """
9
10 class Solution:
11     # @param head: the head of linked list.
12     # @return: a middle node of the linked list
13     def middleNode(self, head):
14         # Write your code here
15         if head is None:
16             return None
17         slow = head;
18         fast = head;
19         while fast.next is not None and fast.next.next is not None:
20             slow = slow.next
21             fast = fast.next.next
22         return slow
```

1.2 Problem ID: 366 Fibonacci

1.2.1 Description

Find the Nth number in Fibonacci sequence.

A Fibonacci sequence is defined as follow:

The first two numbers are 0 and 1.

The i th number is the sum of i-1 th number and i-2 th number.

The first ten numbers in Fibonacci sequence is:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34 ...

1.2.2 Example

Given 1, return 0

Given 2, return 1

Given 10, return 34

1.2.3 Code

C++

```
1 class Solution{
2 public:
3     /**
4      * @param n: an integer
5      * @return an integer f(n)
6      */
7     int fibonacci(int n) {
8         // write your code here
9         n--;
10        double sqrt5=sqrt((double)5);
11        return (pow((1+sqrt5),n)-pow((1-sqrt5),n))/(pow((double)2,n)*sqrt5);
12    }
13};
```

Python

```
1 class Solution:
2     # @param n: an integer
3     # @return an integer f(n)
4     def fibonacci(self, n):
5         # write your code here
6         a = 0;
```

```
7     b = 1;
8     for i in range(n - 1):
9         a, b = b, a + b
10    return a
```

1.3 Problem ID: 452 Remove Linked List Elements

1.3.1 Description

Remove all elements from a linked list of integers that have value val.

1.3.2 Example

Given 1->2->3->3->4->5->3, val = 3, you should return the list as 1->2->4->5

1.3.3 Code

C++

```
1  /**
2   * Definition for singly-linked list.
3   * struct ListNode {
4   *     int val;
5   *     ListNode *next;
6   *     ListNode(int x) : val(x), next(NULL) {}
7   * };
8   */
9  class Solution {
10 public:
11     /**
12      * @param head a ListNode
13      * @param val an integer
14      * @return a ListNode
15      */
16     ListNode *removeElements(ListNode *head, int val) {
17         // Write your code here
18         ListNode dummy;
19         dummy.next = head;
20         head = &dummy;
21         while (head->next != NULL) {
22             if (head->next->val == val) {
23                 head->next = head->next->next;
24             }
25             else {
26                 head = head->next;
27             }
28         }
29         return dummy.next;
30     }
31 };
```

Python

```
1 # Definition for singly-linked list.
2 # class ListNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.next = None
6
7 class Solution:
8     # @param head, a ListNode
9     # @param val, an integer
10    # @return a ListNode
11    def removeElements(self, head, val):
12        # Write your code here
13        if head is None:
14            return head
15        dummy = ListNode(0)
16        dummy.next = head
17        head = dummy
18        while head.next is not None:
19            if head.next.val == val:
20                head.next = head.next.next
21            else:
22                head = head.next
23        return dummy.next
```


第二章 容易 (Easy)

2.1 Problem ID: 1 A + B Problem

2.1.1 Description

Write a function that add two numbers A and B. You should not use + or any arithmetic operators.

2.1.2 Clarification

Are a and b both 32-bit integers?

Yes.

Can I use bit operation?

Sure you can.

2.1.3 Example

Given a=1 and b=2 return 3

2.1.4 Code

C++

```
1  class Solution {
2  public:
3      /*
4       * @param a: The first integer
5       * @param b: The second integer
6       * @return: The sum of a and b
7       */
8      int aplusb(int a, int b) {
9          // write your code here, try to do it without arithmetic operators.
10         if(b == 0){
11             return a;
12         }
13         else{
14             aplusb(a ^ b, (a & b) << 1);
15         }
16     }
17 };
```

Python


```

1  class Solution:
2      """
3      @param a: The first integer
4      @param b: The second integer
5      @return: The sum of a and b
6      """
7
8      def aplusb(self, a, b):
9          # write your code here, try to do it without arithmetic operators.
10         return a + b

```

Note: Python version will be Updated in future.

2.2 Problem ID: 2 Trailing Zeros

2.2.1 Description

Write an algorithm which computes the number of trailing zeros in n factorial.

2.2.2 Example

$11! = 39916800$, so the out should be 2

2.2.3 Code

C++

```

1  class Solution {
2  public:
3      // param n : description of n
4      // return: description of return
5      long long trailingZeros(long long n) {
6          long long sum = 0;
7          while(n!=0){
8              sum += n / 5;
9              n = n / 5;
10         }
11         return sum;
12     }
13 };

```

Python

```
1 class Solution:
2     # @param n a integer
3     # @return ans a integer
4     def trailingZeros(self, n):
5         sum = 0
6         while n != 0:
7             sum += n // 5
8             n = n // 5
9         return sum
```

2.3 Problem ID: 6 Merge Sorted Array II

2.3.1 Description

Merge two given sorted integer array A and B into a new sorted integer array.

2.3.2 Example

```
A=[1,2,3,4]
B=[2,4,5,6]
return [1,2,2,3,4,4,5,6]
```

2.3.3 Code

C++

```
1 class Solution {
2 public:
3     /**
4      * @param A and B: sorted integer array A and B.
5      * @return: A new sorted integer array
6      */
7     vector<int> mergeSortedArray(vector<int> &A, vector<int> &B) {
8         // write your code here
9         vector<int> C;
10        int i = 0, j = 0;
11        while(i < A.size() && j < B.size()){
12            if(A[i] < B[j]){
13                C.push_back(A[i++]);
14            } else {
15                C.push_back(B[j++]);
```

```

16         }
17     }
18     while(i < A.size()){
19         C.push_back(A[i++]);
20     }
21     while(j < B.size()){
22         C.push_back(B[j++]);
23     }
24     return C;
25 }
26 };

```

Python

```

1 class Solution:
2     #@param A and B: sorted integer array A and B.
3     #@return: A new sorted integer array
4     def mergeSortedArray(self, A, B):
5         # write your code here
6         C = []
7         i = 0
8         j = 0
9         while i < len(A) and j < len(B):
10             if A[i] < B[j]:
11                 C.append(A[i])
12                 i = i + 1
13             else:
14                 C.append(B[j])
15                 j = j + 1
16         if i < len(A):
17             C.extend(A[i:])
18         if j < len(B):
19             C.extend(B[j:])
20         return C

```

2.4 Problem ID: 8 Rotate String

2.4.1 Description

Given a string and an offset, rotate string by offset. (rotate from left to right)

2.4.2 Example

Given "abcdefg".

offset=0 => "abcdefg"

offset=1 => "gabcdef"

offset=2 => "fgabcde"

offset=3 => "efgabcd"

2.4.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param str: a string
5       * @param offset: an integer
6       * @return: nothing
7       */
8      void rotateString(string &str, int offset){
9          //wirte your code here
10         if (str.size() == 0){
11
12         }else{
13             offset = offset % str.size();
14             str = str.substr(str.size() - offset, offset) +
15                 str.substr(0, str.size() - offset);
16         }
17     }
18 };
```

Python

```
1  class Solution:
2      # @param s: a list of char
3      # @param offset: an integer
4      # @return: nothing
5      def rotateString(self, s, offset):
6          # write you code here
7          if len(s) != 0:
8              offset = offset % len(s)
9              s[:] = s[-offset:] + s[:-offset]
```

2.5 Problem ID: 9 Fizz Buzz

2.5.1 Description

Given number n. Print number from 1 to n. But:
when number is divided by 3, print "fizz".

when number is divided by 5, print "buzz".

when number is divided by both 3 and 5, print "fizz buzz".

2.5.2 Example

If $n = 15$, you should return:

```
[  
  "1", "2", "fizz",  
  "4", "buzz", "fizz",  
  "7", "8", "fizz",  
  "buzz", "11", "fizz",  
  "13", "14", "fizz buzz"  
]
```

2.5.3 Code

C++

```
1  class Solution {  
2  public:  
3      /**  
4       * param n: As description.  
5       * return: A list of strings.  
6       */  
7      vector<string> fizzBuzz(int n) {  
8          vector<string> results;  
9          for (int i = 1; i <= n; i++) {  
10             if (i % 15 == 0) {  
11                 results.push_back("fizz_buzz");  
12             } else if (i % 5 == 0) {  
13                 results.push_back("buzz");  
14             } else if (i % 3 == 0) {  
15                 results.push_back("fizz");  
16             } else {  
17                 results.push_back(to_string(i));  
18             }  
19         }  
20         return results;  
21     }  
22 };
```

Python

```

1 class Solution:
2     """
3     @param n: An integer as description
4     @return: A list of strings.
5     For example, if n = 7, your code should return
6         ["1", "2", "fizz", "4", "buzz", "fizz", "7"]
7     """
8     def fizzBuzz(self, n):
9         results = []
10        for i in range(1, n+1):
11            if i % 15 == 0:
12                results.append("fizz_buzz")
13            elif i % 5 == 0:
14                results.append("buzz")
15            elif i % 3 == 0:
16                results.append("fizz")
17            else:
18                results.append(str(i))
19        return results

```

2.6 Problem ID: 13 strStr

2.6.1 Description

For a given source string and a target string, you should output the first index(from 0) of target string in source string.

If target does not exist in source, just return -1.

2.6.2 Clarification

Do I need to implement KMP Algorithm in a real interview?

Not necessary. When you meet this problem in a real interview, the interviewer may just want to test your basic implementation ability. But make sure your confirm with the interviewer first.

2.6.3 Example

If source = "source" and target = "target", return -1.

If source = "abcdabcdefg" and target = "bcd", return 1.

2.6.4 Code

C++

```
1  #include <cstring>
2  #include <iostream>
3  using namespace std;
4  class Solution {
5  public:
6      /**
7       * Returns a index to the first occurrence of target in source,
8       * or -1 if target is not part of source.
9       * @param source string to be scanned.
10      * @param target string containing the sequence of characters to match.
11      */
12     int strStr(const char *source, const char *target) {
13         // write your code here
14         if (source == NULL || target == NULL){
15             return -1;
16         }
17         int size_source = strlen(source);
18         int size_target = strlen(target);
19         int i, j;
20         for (i = 0; i < size_source - size_target + 1; i++){
21             for(j = 0; j < size_target; j++){
22                 if(source[i + j] != target[j]){
23                     break;
24                 }
25             }
26             if(j == size_target){
27                 return i;
28             }
29         }
30         return -1;
31     }
32 };
```

Python

```
1  class Solution:
2      def strStr(self, source, target):
3          # write your code here
4          if source is None or target is None:
5              return -1
6          return source.find(target)
```

2.7 Problem ID: 14 First Position of Target

2.7.1 Description

For a given sorted array (ascending order) and a target number, find the first index of this number in $O(\log n)$ time complexity.

If the target number does not exist in the array, return -1.

2.7.2 Example

If the array is [1, 2, 3, 3, 4, 5, 10], for given target 3, return 2.

2.7.3 Code

C++

```
1  #include <cstring>
2  #include <iostream>
3  using namespace std;
4  class Solution {
5  public:
6      /**
7       * Returns a index to the first occurrence of target in source,
8       * or -1 if target is not part of source.
9       * @param source string to be scanned.
10      * @param target string containing the sequence of characters to match.
11      */
12     int strStr(const char *source, const char *target) {
13         // write your code here
14         if (source == NULL || target == NULL){
15             return -1;
16         }
17         int size_source = strlen(source);
18         int size_target = strlen(target);
19         int i, j;
20         for (i = 0; i < size_source - size_target + 1; i++){
21             for(j = 0; j < size_target; j++){
22                 if(source[i + j] != target[j]){
23                     break;
24                 }
25             }
26             if(j == size_target){
27                 return i;
28             }
29         }
30         return -1;
31     }
32 };
```


Python

```
1 class Solution:
2     # @param nums: The integer array
3     # @param target: Target number to find
4     # @return the first position of target in nums, position start from 0
5     def binarySearch(self, nums, target):
6         # write your code here
7         start = 0
8         end = len(nums) - 1
9         while start + 1 < end:
10             mid = (start + end) >> 1
11             if nums[mid] < target:
12                 start = mid
13             else:
14                 end = mid
15         if nums[start] == target:
16             return start
17         if nums[end] == target:
18             return end
19         return -1
```

2.8 Problem ID: 28 Search a 2D Matrix

2.8.1 Description

Write an efficient algorithm that searches for a value in an $m \times n$ matrix.

This matrix has the following properties:

Integers in each row are sorted from left to right.

The first integer of each row is greater than the last integer of the previous row.

2.8.2 Example

Consider the following matrix:

```
[
  [1, 3, 5, 7],
  [10, 11, 16, 20],
  [23, 30, 34, 50]
]
```

Given target = 3, return true.

2.8.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param matrix, a list of lists of integers
5       * @param target, an integer
6       * @return a boolean, indicate whether matrix contains target
7       */
8      bool searchMatrix(vector<vector<int>> &matrix, int target) {
9          // write your code here
10         int n = matrix.size();
11         if (n == 0){
12             return false;
13         }
14         int m = matrix[0].size();
15         if (m == 0){
16             return false;
17         }
18
19         int start = 0;
20         int end = m * n - 1;
21         int mid, row, col;
22
23         while (start + 1 < end){
24             mid = start + (end - start) / 2;
25             row = mid / m;
26             col = mid % m;
27             if (matrix[row][col] == target){
28                 return true;
29             }else if(matrix[row][col] < target){
30                 start = mid;
31             }else{
32                 end = mid;
33             }
34         }
35         if (matrix[start / m][start % m] == target){
36             return true;
37         }
38         if (matrix[end / m][end % m] == target){
39             return true;
40         }
41         return false;
42     }
43 };
```

Python

```

1  class Solution:
2      """
3      @param matrix, a list of lists of integers
4      @param target, an integer
5      @return a boolean, indicate whether matrix contains target
6      """
7      def searchMatrix(self, matrix, target):
8          # write your code here
9          m = len(matrix)
10         if m == 0:
11             return False
12         n = len(matrix[0])
13         if n == 0:
14             return False
15         start = 0
16         end = m * n - 1
17         while start + 1 < end:
18             mid = (start + end) / 2
19             x = mid / n
20             y = mid % n
21             if matrix[x][y] < target:
22                 start = mid
23             else:
24                 end = mid
25         x, y = start / n, start % n
26         if matrix[x][y] == target:
27             return True
28         x, y = end / n, end % n
29         if matrix[x][y] == target:
30             return True
31         return False

```

2.9 Problem ID: 82 Single Number**2.9.1 Description**

Given $2*n + 1$ numbers, every numbers occurs twice except one, find it.

2.9.2 Example

Given $[1,2,2,1,3,4,3]$, return 4

2.9.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param A: Array of integers.
5       * return: The single number.
6       */
7      int singleNumber(vector<int> &A) {
8          // write your code here
9          int x;
10         for(int i = 0; i < A.size(); i++){
11             x ^= A[i];
12         }
13         return x;
14     }
15 };
```

Python

```
1  class Solution:
2      """
3      @param A : an integer array
4      @return : a integer
5      """
6      def singleNumber(self, A):
7          # write your code here
8          x = 0
9          for a in A:
10             x ^= a
11         return x
```

2.10 Problem ID: 157 Unique Characters

2.10.1 Description

Implement an algorithm to determine if a string has all unique characters.

2.10.2 Example

Given "abc", return true.

Given "aab", return false.

2.10.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param str: a string
5       * @return: a boolean
6       */
7      bool isUnique(string &str) {
8          // write your code here
9          int ch[128] = {0};
10         for(int i = 0; i < str.length(); i++){
11             if(ch[str[i]] != 0){
12                 return false;
13             }else{
14                 ch[str[i]] = 1;
15             }
16         }
17         return true;
18     }
19 };
```

Python

```
1  class Solution:
2      # @param s: a string
3      # @return: a boolean
4      def isUnique(self, str):
5          # write your code here
6          ch = range(129)
7          for i in range(129):
8              ch[i] = 0
9          for s in str:
10             if ch[ord(s)] != 0:
11                 return False
12             ch[ord(s)] = ch[ord(s)] + 1
13         return True
```

2.11 Problem ID: 166 Nth to Last Node in List

2.11.1 Description

Find the nth to last element of a singly linked list.

The minimum number of nodes in list is n.

2.11.2 Example

Given a List 3->2->1->5->null and n = 2, return node whose value is 1.

2.11.3 Code

C++

```

1  /**
2   * Definition of ListNode
3   * class ListNode {
4   * public:
5   *     int val;
6   *     ListNode *next;
7   *     ListNode(int val) {
8   *         this->val = val;
9   *         this->next = NULL;
10  *     }
11  * }
12  */
13 class Solution {
14 public:
15     /**
16      * @param head: The first node of linked list.
17      * @param n: An integer.
18      * @return: Nth to last node of a singly linked list.
19      */
20     ListNode *nthToLast(ListNode *head, int n) {
21         // write your code here
22         if(head == NULL || n < 1){
23             return NULL;
24         }
25         ListNode *p1 = head;
26         ListNode *p2 = head;
27         for(int i = 0; i < n; i++){
28             if(p2 != NULL){
29                 p2 = p2->next;
30             }else{
31                 return NULL;
32             }
33         }
34         while(p2 != NULL){
35             p1 = p1->next;
36             p2 = p2->next;
37         }
38         return p1;
39     }

```

```
40 };
```

Python

```

1  ?? ?? ??
2  Definition of ListNode
3  class ListNode(object):
4
5      def __init__(self, val, next=None):
6          self.val = val
7          self.next = next
8  ?? ?? ??
9  class Solution:
10     ?? ?? ??
11     @param head: The first node of linked list.
12     @param n: An integer.
13     @return: Nth to last node of a singly linked list.
14     ?? ?? ??
15     def nthToLast(self, head, n):
16         # write your code here
17         if head is None or n < 1:
18             return None
19         p1 = head
20         p2 = head
21         for i in range(0,n):
22             if p2 is not None:
23                 p2 = p2.next
24             else:
25                 return None
26         while p2 is not None:
27             p2 = p2.next
28             p1 = p1.next
29         return p1

```

2.12 Problem ID: 204 Singleton

2.12.1 Description

Singleton is a most widely used design pattern. If a class has and only has one instance at every moment, we call this design as singleton. For example, for class Mouse (not a animal mouse), we should design it in singleton.

Your job is to implement a getInstance method for given class, return the same instance of this class every time you call this method.

2.12.2 Example

n Java:

```
A a = A.getInstance();
```

```
A b = A.getInstance();
```

a should equal to b.

2.12.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @return: The same instance of this class every time
5       */
6      static Solution* getInstance() {
7          // write your code here
8          static Solution * m_pInstance;
9          if(m_pInstance == NULL){
10             m_pInstance = new Solution();
11          }
12          return m_pInstance;
13      }
14  private:
15      Solution(){
16
17      }
18  };
```

Python

```
1  class Solution:
2      # @return: The same instance of this class every time
3      instance = None
4      @classmethod
5      def getInstance(cls):
6          # write your code here
7          if cls.instance is None:
8              cls.instance = Solution()
9          return cls.instance
```


2.13 Problem ID: 496 Toy Factory

2.13.1 Description

Factory is a design pattern in common usage. Please implement a ToyFactory which can generate proper toy based on the given type.

2.13.2 Example

```
1 ToyFactory tf = ToyFactory();
2 Toy toy = tf.getToy('Dog');
3 toy.talk();
4 >> Wow
5
6 toy = tf.getToy('Cat');
7 toy.talk();
8 >> Meow
```

2.13.3 Code

C++

```
1  /**
2   * Your object will be instantiated and called as such:
3   * ToyFactory* tf = new ToyFactory();
4   * Toy* toy = tf->getToy(type);
5   * toy->talk();
6   */
7  class Toy {
8  public:
9      virtual void talk() const=0;
10 };
11
12 class Dog: public Toy {
13     // Write your code here
14     void talk() const{
15         cout << "Wow" << endl;
16     }
17 };
18
19 class Cat: public Toy {
20     // Write your code here
21     void talk() const{
22         cout << "Meow" << endl;
23     }
24 };
25
26 class ToyFactory {
```

```

27 public:
28     /**
29      * @param type a string
30      * @return Get object of the type
31      */
32     Toy* getToy(string& type) {
33         // Write your code here
34         if(type == "Dog"){
35             return new Dog();
36         }
37         if(type == "Cat"){
38             return new Cat();
39         }
40         return NULL;
41     }
42 };

```

Python

```

1  """
2  Your object will be instantiated and called as such:
3  ty = ToyFactory()
4  toy = ty.getToy(type)
5  toy.talk()
6  """
7  class Toy:
8      def talk(self):
9          raise NotImplementedError('This method should have implemented.')
10
11  class Dog(Toy):
12      # Write your code here
13      def talk(self):
14          print "Wow"
15
16  class Cat(Toy):
17      # Write your code here
18      def talk(self):
19          print "Meow"
20
21
22  class ToyFactory:
23      # @param {string} shapeType a string
24      # @return {Toy} Get object of the type
25      def getToy(self, type):
26          # Write your code here
27          if type == "Dog":
28              return Dog()
29          if type == "Cat":
30              return Cat()
31          return None

```

2.14 Problem ID: 497 Shape Factory

2.14.1 Description

Factory is design pattern in common usage. Implement a ShapeFactory that can generate correct shape.

2.14.2 Example

```

1 ShapeFactory sf = new ShapeFactory();
2 Shape shape = sf.getShape("Square");
3 shape.draw();
4 >>  ____
5 >> |    |
6 >> |    |
7 >>  ____
8
9 shape = sf.getShape("Triangle");
10 shape.draw();
11 >>  /\
12 >> /  \
13 >> /____\
14
15 shape = sf.getShape("Rectangle");
16 shape.draw();
17 >>  ____
18 >> |    |
19 >>  ____

```

2.14.3 Code

C++

```

1 /**
2  * Your object will be instantiated and called as such:
3  * ShapeFactory* sf = new ShapeFactory();
4  * Shape* shape = sf->getShape(shapeType);
5  * shape->draw();
6  */
7 class Shape {
8 public:
9     virtual void draw() const=0;
10 };
11
12 class Rectangle: public Shape {
13     // Write your code here
14     void draw() const{
15         cout << "____" << endl << "|    |" << endl << "____" << endl;

```

```

16     }
17 };
18
19 class Square: public Shape {
20     // Write your code here
21     void draw() const{
22         cout << "┐───" << endl << "│││││" << endl
23         << "│││││" << endl << "└───" << endl;
24     }
25 };
26
27 class Triangle: public Shape {
28     // Write your code here
29     void draw() const{
30         cout << "┐┐/\\" << endl << "┐/┐┐\\" << endl << "/____\\" << endl;
31     }
32 };
33
34 class ShapeFactory {
35 public:
36     /**
37      * @param shapeType a string
38      * @return Get object of type Shape
39      */
40     Shape* getShape(string& shapeType) {
41         // Write your code here
42         if(shapeType == "Square"){
43             return new Square();
44         }
45         if(shapeType == "Rectangle"){
46             return new Rectangle();
47         }
48         if(shapeType == "Triangle"){
49             return new Triangle();
50         }
51         return NULL;
52     }
53 };

```

Python

```

1  """
2  Your object will be instantiated and called as such:
3  sf = ShapeFactory()
4  shape = sf.getShape(shapeType)
5  shape.draw()
6  """
7  class Shape:
8      def draw(self):
9          raise NotImplementedError('This method should have implemented.')
10
11  class Triangle(Shape):

```

```
12     # Write your code here.
13     def draw(self):
14         print "uu/\\"
15         print "u/uu\\"
16         print "/____\\"
17
18     class Rectangle(Shape):
19         # Write your code here
20         def draw(self):
21             print "└───"
22             print "|uuu|"
23             print "└───"
24
25     class Square(Shape):
26         # Write your code here
27         def draw(self):
28             print "└───"
29             print "|uuu|"
30             print "|uuu|"
31             print "└───"
32
33     class ShapeFactory:
34         # @param {string} shapeType a string
35         # @return {Shape} Get object of type Shape
36         def getShape(self, shapeType):
37             # Write your code here
38             if shapeType == "Square":
39                 return Square()
40             if shapeType == "Triangle":
41                 return Triangle()
42             if shapeType == "Rectangle":
43                 return Rectangle()
44             return None
```


第三章 中等 (Medium)

3.1 Problem ID: 3 Digit Counts

3.1.1 Description

Count the number of k's between 0 and n. k can be 0 - 9.

3.1.2 Example

if n=12, k=1 in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], we have FIVE 1's (1, 10, 11, 12)

3.1.3 Code

C++

```
1 class Solution {
2 public:
3     /*
4      * param k : As description.
5      * param n : As description.
6      * return: How many k's between 0 and n.
7      */
8     int digitCounts(int k, int n) {
9         // write your code here
10        int count = 0;
11        for(int i = 0; i <= n; i++){
12            int j = i;
13            while(true){
14                if(j % 10 == k){
15                    count++;
16                }
17                j = j / 10;
18                if(j == 0){
19                    break;
20                }
21            }
22        }
23        return count;
24    }
25};
```

Python

```
1 class Solution:
2     # @param k & n two integer
3     # @return ans a integer
4     def digitCounts(self, k, n):
```



```
5     assert(n >= 0 and 0 <= k <= 9)
6     count = 0
7     for i in range(n + 1):
8         j = i
9         while True:
10             if j % 10 == k:
11                 count = count + 1
12             j = j // 10
13             if j == 0:
14                 break
15     return count
```

3.2 Problem ID: 4 Ugly Number II

3.2.1 Description

Ugly number is a number that only have factors 2, 3 and 5.

Design an algorithm to find the nth ugly number. The first 10 ugly numbers are 1, 2, 3, 4, 5, 6, 8, 9, 10, 12...

Notice Note that 1 is typically treated as an ugly number.

3.2.2 Example

If n=9, return 10.

3.2.3 Code

C++

```
1  class Solution {
2  public:
3      /*
4       * @param n an integer
5       * @return the nth prime number as description.
6       */
7      int nthUglyNumber(int n) {
8          // write your code here
9          int *uglys = new int[n];
10         uglys[0] = 1;
11         int next = 1;
12         int *p2 = uglys;
13         int *p3 = uglys;
14         int *p5 = uglys;
15         while(next < n){
16             int m = min(min(*p2 * 2, *p3 * 3), *p5 * 5);
17             uglys[next] = m;
18             while(*p2 * 2 <= uglys[next]){
19                 p2++;
20             }
21             while(*p3 * 3 <= uglys[next]){
22                 p3++;
23             }
24             while(*p5 * 5 <= uglys[next]){
25                 p5++;
26             }
27             next++;
28         }
29         int uglyNum = uglys[n - 1];
30         delete[] uglys;
```

```
31         return uglyNum;
32     }
33 };
```

Python

```
1  class Solution:
2      """
3      @param {int} n an integer.
4      @return {int} the nth prime number as description.
5      """
6      def nthUglyNumber(self, n):
7          # write your code here
8          ugllys = []
9          ugllys.append(1)
10         p2 = 0
11         p3 = 0
12         p5 = 0
13         next = 1
14         while next < n:
15             m = min(ugllys[p2] * 2, ugllys[p3] * 3, ugllys[p5] * 5)
16             ugllys.append(m)
17             while ugllys[p2] * 2 <= ugllys[next]:
18                 p2 = p2 + 1
19             while ugllys[p3] * 3 <= ugllys[next]:
20                 p3 = p3 + 1
21             while ugllys[p5] * 5 <= ugllys[next]:
22                 p5 = p5 + 1
23             next = next + 1
24         return ugllys[n - 1]
```

3.3 Problem ID: 5 Kth Largest Element

3.3.1 Description

Find K-th largest element in an array.

3.3.2 Example

In array [9,3,2,4,8], the 3rd largest element is 4.

In array [1,2,3,4,5], the 1st largest element is 5, 2nd largest element is 4, 3rd largest element is 3 and etc.

3.3.3 Code

C++

```
1 class Solution {
2 public:
3     /*
4      * param k : description of k
5      * param nums : description of array and index 0 ~ n-1
6      * return: description of return
7      */
8     int kthLargestElement(int k, vector<int> nums) {
9         // write your code here
10        sort(nums.rbegin(), nums.rend());
11        return nums[k-1];
12    }
13};
```

Python

```
1 class Solution:
2     # @param k & A a integer and an array
3     # @return ans a integer
4     def kthLargestElement(self, k, A):
5         A = sorted(A, reverse = True)
6         return A[k-1]
```

Note: This problem will be update soon!

3.4 Problem ID: 83 Single Number II

3.4.1 Description

Given $3*n + 1$ numbers, every numbers occurs triple times except one, find it.

3.4.2 Example

Given $[1,1,2,3,3,3,2,2,4,1]$ return 4

3.4.3 Code

C++

```

1  class Solution {
2  public:
3      /**
4       * @param A : An integer array
5       * @return : An integer
6       */
7      int singleNumberII(vector<int> &A) {
8          // write your code here
9          int bitnum[32] = {0};
10         int res = 0;
11         for(int i = 0; i < 32; i++){
12             for(int j = 0; j < A.size(); j++){
13                 bitnum[i] += (A[j] >> i) & 1;
14             }
15             res |= (bitnum[i] % 3) << i;
16         }
17         return res;
18     }
19 };

```

Python

```

1  class Solution:
2      """
3      @param A : An integer array
4      @return : An integer
5      """
6      def singleNumberII(self, A):
7          # write your code here
8          res = 0
9          bitnum = [0 for i in range(0,32)]
10         for i in range(0,32):

```

```
11         for j in range(0, len(A)):  
12             bitnum[i] += (A[j] >> i) & 1  
13             res |= (bitnum[i] % 3) << i  
14         return res
```

3.5 Problem ID: 84 Single Number III

3.5.1 Description

Given $2*n + 2$ numbers, every numbers occurs twice except two, find them.

3.5.2 Example

Given $[1,2,2,3,4,4,5,3]$ return 1 and 5

3.5.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param A : An integer array
5       * @return : Two integers
6       */
7      vector<int> singleNumberIII(vector<int> &A) {
8          // write your code here
9          int Xor = 0;
10         for(int i = 0; i < A.size(); i++){
11             Xor ^= A[i];
12         }
13         int lowbit = Xor & -Xor;
14         int a = 0;
15         int b = 0;
16         for(int i = 0; i < A.size(); i++){
17             if(A[i] & lowbit){
18                 a ^= A[i];
19             }else{
20                 b ^= A[i];
21             }
22         }
23         vector<int> ans;
24         ans.push_back(a);
25         ans.push_back(b);
26         return ans;
27     }
28 };
```

Python

```
1 class Solution:
2     """
3     @param A : An integer array
4     @return : Two integer
5     """
6     def singleNumberIII(self, A):
7         # write your code here
8         xor = reduce(lambda x, y : x ^ y, A)
9         lowbit = xor & -xor
10        a, b = 0, 0
11        for num in A:
12            if num & lowbit:
13                a ^= num
14            else:
15                b ^= num
16        return (a, b)
```


3.6 Problem ID: 162 Set Matrix Zeroes

3.6.1 Description

Given a $m \times n$ matrix, if an element is 0, set its entire row and column to 0. Do it in place.

3.6.2 Example

Given a matrix

```
[
  [1,2],
  [0,3]
],
return
[
  [0,2],
  [0,0]
]
```

3.6.3 Code

C++

```
1 class Solution {
2 public:
3     /**
4      * @param matrix: A list of lists of integers
5      * @return: Void
6      */
7     void setZeroes(vector<vector<int>> &matrix) {
8         // write your code here
9         if(matrix.size() == 0){
10             return ;
11         }
12         int m = matrix.size();
13         int n = matrix[0].size();
14         bool* row = new bool[m];
15         bool* col = new bool[n];
16         for(int i = 0; i < m; i++){
17             row[i] = false;
18         }
19         for(int i = 0; i < n; i++){
```

```

20         col[i] = false;
21     }
22     for(int i = 0; i < m; i++){
23         for(int j = 0; j < n; j++){
24             if(matrix[i][j] == 0){
25                 row[i] = true;
26                 col[j] = true;
27             }
28         }
29     }
30     for(int i = 0; i < m; i++){
31         for(int j = 0; j < n; j++){
32             if(row[i] == true || col[j] == true){
33                 matrix[i][j] = 0;
34             }
35         }
36     }
37 }
38 };

```

Python

```

1  class Solution:
2      """
3      @param matrix: A list of lists of integers
4      @return: Nothing
5      """
6      def setZeroes(self, matrix):
7          # write your code here
8          if len(matrix) == 0:
9              return
10         rownum, colnum = len(matrix), len(matrix[0])
11         row = [False for i in range(rownum)]
12         col = [False for i in range(colnum)]
13         for i in range(rownum):
14             for j in range(colnum):
15                 if matrix[i][j] == 0:
16                     row[i], col[j] = True, True
17         for i in range(rownum):
18             for j in range(colnum):
19                 if row[i] == True or col[j] == True:
20                     matrix[i][j] = 0

```

3.7 Problem ID: 394 Coins in a Line

3.7.1 Description

There are n coins in a line. Two players take turns to take one or two coins from right side until there are no more coins left. The player who take the last coin wins.

Could you please decide the first play will win or lose?

3.7.2 Example

$n = 1$, return true.

$n = 2$, return true.

$n = 3$, return false.

$n = 4$, return true.

$n = 5$, return true.

3.7.3 Code

C++

```
1 class Solution {
2 public:
3     /**
4      * @param n: an integer
5      * @return: a boolean which equals to true if the first player will win
6      */
7     bool firstWillWin(int n) {
8         // write your code here
9         if(n % 3 == 0){
10             return false;
11         }
12         return true;
13     }
14 };
```

Python

```
1 class Solution:
2     # @param n: an integer
3     # @return: a boolean which equals to True if the first player will win
4     def firstWillWin(self, n):
5         # write your code here
```

```
6     if n % 3 == 0:  
7         return False  
8     return True
```

3.8 Problem ID: 419 Roman to Integer

3.8.1 Description

Given a roman numeral, convert it to an integer.

The answer is guaranteed to be within the range from 1 to 3999.

3.8.2 Clarification

What is Roman Numeral?

https://en.wikipedia.org/wiki/Roman_numerals

<https://zh.wikipedia.org/wiki/%E7%BD%97%E9%A9%AC%E6%95%B0%E5%AD%97>

<http://baike.baidu.com/view/42061.htm>

3.8.3 Example

IV -> 4

XII -> 12

XXI -> 21

XCIX -> 99

3.8.4 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param s Roman representation
5       * @return an integer
6       */
7      int romanToInt(string& s) {
8          // Write your code here
9          map<char, int> num;
10         num['I'] = 1;
11         num['V'] = 5;
12         num['X'] = 10;
13         num['L'] = 50;
14         num['C'] = 100;
15         num['D'] = 500;
16         num['M'] = 1000;
17         int ans = num[s[s.length() - 1]];
18         for (int i = s.length() - 2; i >= 0; i--) {
```

```
19         if (num[s[i]] < num[s[i + 1]]){
20             ans -= num[s[i]];
21         }
22         else{
23             ans += num[s[i]];
24         }
25     }
26     return ans;
27 }
28 };
```

Python

```
1 class Solution:
2     # @param {string} s Roman representation
3     # @return {int} an integer
4     def romanToInt(self, s):
5         # Write your code here
6         if s == "":
7             return 0
8         roman = {'I':1, 'V':5, 'X':10, 'L':50, 'C':100, 'D':500, 'M':1000}
9         sum = roman[s[-1]]
10        i = len(s) - 2
11        while i >= 0:
12            if roman[s[i]] < roman[s[i + 1]]:
13                sum -= roman[s[i]]
14            else:
15                sum += roman[s[i]]
16            i -= 1
17        return sum
```

3.9 Problem ID: 428 Pow(x, n)

3.9.1 Description

Implement $\text{pow}(x, n)$.

3.9.2 Example

$\text{Pow}(2.1, 3) = 9.261$

$\text{Pow}(0, 1) = 0$

$\text{Pow}(1, 0) = 1$

3.9.3 Code

C++

```
1  class Solution {
2  public:
3      /**
4       * @param x the base number
5       * @param n the power number
6       * @return the result
7       */
8      double myPow(double x, int n) {
9          // Write your code here
10         if(n < 0){
11             return 1.0 / myPow(x, -n);
12         }
13         if(n == 0){
14             return 1;
15         }
16         if(n % 2 == 0){
17             return myPow(x, n>>1) * myPow(x, n>>1);
18         }else{
19             return myPow(x, n>>1) * myPow(x, n>>1) * x;
20         }
21     }
22 };
```

Python

```
1  class Solution:
2      # @param {double} x the base number
3      # @param {int} n the power number
4      # @return {double} the result
5      def myPow(self, x, n):
```

```
6      # Write your code here
7      if n < 0:
8          return 1.0 / self.myPow(x, -n)
9      if n == 0:
10         return 1
11     if n % 2 == 0:
12         return self.myPow(x, n >> 1) * self.myPow(x, n >> 1)
13     else:
14         return self.myPow(x, n >> 1) * self.myPow(x, n >> 1) * x
```


第四章 困难 (Hard)

第五章 超难 (Super)