

LintCode 参考程序

Lionx

2016 年 4 月 20 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: 496 Toy Factory	15
2.2.1 Description	15
2.2.2 Example	15
2.2.3 Code	15

2.3	Problem ID: 497 Shape Factory	17
2.3.1	Description	17
2.3.2	Example	17
2.3.3	Code	18
第三章	中等 (Medium)	21
3.1	Problem ID: 394 Coins in a Line	22
3.1.1	Description	22
3.1.2	Example	22
3.1.3	Code	22
3.2	Problem ID: 428 Pow(x, n)	24
3.2.1	Description	24
3.2.2	Example	24
3.2.3	Code	24
第四章	困难 (Hard)	27
第五章	超难 (Super)	29

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: 496 Toy Factory

2.2.1 Description

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

2.2.2 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

```

2.2.3 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

```

2.3 Problem ID: 497 Shape Factory

2.3.1 Description

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

2.3.2 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 >>  ——

```

2.3.3 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 "  /\\"
15          print "/  /\\"
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: 394 Coins in a Line

3.1.1 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?

3.1.2 Example

$n = 1$, return true.

$n = 2$, return true.

$n = 3$, return false.

$n = 4$, return true.

$n = 5$, return true.

3.1.3 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
```

3.2 Problem ID: 428 Pow(x, n)

3.2.1 Description

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

3.2.2 Example

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

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

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

3.2.3 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)