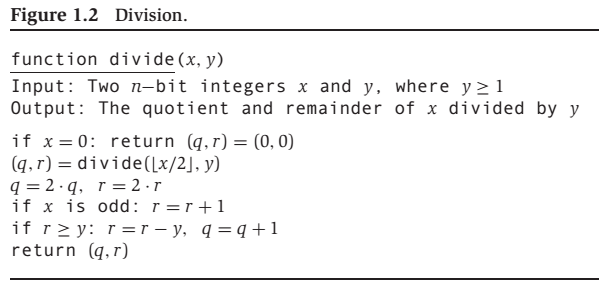
# 算法设计与应用基础: Homework No 1

## Deadline: April 18, 2018 (Wednesday)

Note: Please label your work with homeworkID \_student ID\_name(eg, 1\_1433XXXX\_张三). In B. Programming, ‘#num’ is the number of problem in leetcode. You can do your homework in leetcode and send it to [algo2017@163.com](mailto:algo2017@163.com) when you submit all the solutions correctly.

1. **Excise**
2. Justify the correctness of the following algorithm, and show the time complexity on *n*-bit inputs.



1. Consider the problem of computing *xy* for given integers *x* and *y*: we want the whole answer, not modulo a third number. We know two algorithms for doing this: the iterative algorithm which performs *y*-1 multiplications by *x* and the recursive algorithm based on the binary expansion of *y*.

Compare the time requirements for these two algorithms, assuming the time to multiply *n*-bit number by an *m*-bit number is O(*mn*).

1. In an RSA Cryptosystem, *p*=7 and *q*=11, find appropriate exponents *d* and *e*.
2. **Programming**

#### Remove Nth Node From End of List (#19)

Given a linked list, remove the *n*th node from the end of list and return its head.

For example,

Given linked list: **1->2->3->4->5**, and ***n* = 2**.

After removing the second node from the end, the linked list becomes **1->2->3->5**.

**Note:** Given *n* will always be valid. Try to do this in one pass.

#### Merge Two Sorted Lists (#21)

Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists.

#### Search Insert Position (#35)

Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order. You may assume no duplicates in the array. Here are few examples.

[1,3,5,6], 5 → 2

[1,3,5,6], 2 → 1

[1,3,5,6], 7 → 4

[1,3,5,6], 0 → 0

#### Climbing Stairs (#70)

You are climbing a stair case. It takes *n* steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

#### Same Tree (#100)

Given two binary trees, write a function to check if they are equal or not. Two binary trees are considered equal if they are structurally identical and the nodes have the same value.