Algorithm Analysis

Not Strong Enough

March 7, 2020

1 Bit Complexity of Euclid Algorithm

1.1 Exercise 1

prove the more precise bound of the school method

Assume a has n bits and b has k bits($k < n$), in the school method of division, calculate in rounds.

In each round, extend b by filling zeros into lower bits and make b have same bit size as a. If the extended b is smaller than a, let a minus b.

It is obvious that the first bit of b is 1, so each minus will decrease the bit size of a. Besides, since the minus is made only when the extended b is smaller than the current value of a, if the bit size of a is smaller than k, the calculation will end. So there are at most $n - k$ rounds.

Now consider the minus. Since the lower bits of extended b are 0, only the first k bits of extended b costs. So each minus has k operations.

In all, there are at most $(n - k) * k$ operations, so the complexity is $O(k(n - k))$.

1.2 Exercise 2

prove the complexity of Euclid algorithm is $O(n^2)$

In each round, the Euclid algorithm calculates $a\%b$(assume $a \geq b$) which is less than b. If the result is not 0, it uses the result with b to do the new round. So we can assume that, in each round, the two calculated numbers are:

$$(x_0, x_1), (x_1, x_2) \dots (x_{m-1}, x_m)$$

Where $x_0 > x_1 > \dots x_m, x_{m-1}\%x_m = 0$. Let the bit size of x_i be t_i, then $t_0 = n, t_1 = k$, using the result of Exercise 1 we can find that the total operation number is:

$$O(t_1(t_0 - t_1)) + O(t_2(t_1 - t_2)) + \cdots + O(t_m(t_{m-1} - t_m)) \leq \sum_{i=1}^{m} t_i t_{i-1} - \sum_{i=1}^{m} t_i^2 \tag{1}$$

$$< \sum_{i=0}^{m-1} t_i^2 - \sum_{i=1}^{m} t_i^2 \tag{2}$$

$$= t_0^2 = n^2 \tag{3}$$

So we can see the operation number of Euclid algorithm is $O(n^2)$