International Journal of Trend in Research and Development, Volume 8(5), ISSN: 2394-9333
www.ijtrd.com

Analysis and Optimization of Enumeration Method in Ancient Classical Problem

Wang Juhui,
School of computer science and Technology, Shandong University of Technology, Shandong Zibo, China

Abstract: Through the analysis and optimization of the use of enumeration method in the ancient classical problem "100 money to buy 100 chickens", this paper provides an interesting model for middle school students and college students learning enumeration method, and provides guidance for the complexity analysis of the algorithm, and forms a good idea to solve this kind of problem.

Keywords: *Enumeration, Time Complexity, Optimization*

I. OVERVIEW OF ENUMERATION METHODS

Enumeration method is a common method to solve problems in computer programming. It can be considered to use this method when certain preconditions are met and there are no other problem-solving methods. It belongs to a violent problem-solving method.

Enumeration method is also called exhaustive method. It is understood simply from the literal meaning that all possible solutions of the problem are listed one by one through circulation according to the nature of the problem itself (can not be omitted or repeated), and in the process of enumerating one by one, check whether each possible solution is the real solution of the problem. If so, we adopt this solution, otherwise abandon it.

II. THE BASIC IDEA OF ENUMERATING PROBLEM SOLVING

The key to solving problems by enumeration method: list all possible solutions without omission or repetition, test each possible solution and optimize it within the range of running time requirements. The basic idea of solving problems by enumeration method:

1. Determine the enumeration object.
2. Specify the value range of enumeration objects.
3. Write the relevant conditional expressions according to the requirements of the topic. Here, the conditional expression can be a mathematical expression, a relational expression or a logical expression.
4. The possible solutions are listed by using loop statements to verify whether various conditional expressions are satisfied in the loop body.
5. Optimize the program according to the problem background, so as to narrow the search scope and reduce the running time of the program.

The following analyzes the general idea of enumeration problem-solving with the famous "100 money to buy 100 chickens" in ancient China. This problem is both classic and interesting. It is the first choice for beginners to learn enumeration methods. Problem description [1]: about the 5th century A.D., mathematician Zhang Qiujian put forward a well-known problem of 100 money and 100 chickens in his Suanjing: one cock is worth five, one hen is worth three, three chicks is worth one, 100 money buys 100 chickens, how about cock, hen and chick?
Analysis: this problem is a typical multi solution problem, which needs to "gather" answers. This is an important prerequisite for using enumeration.

III. USING TRIPLE LOOP ENUMERATION

First of all, the easiest thing to think of is to use a triple cycle to list the values of cocks, hens and chicks respectively, and then multiply the number of chickens of each type by the unit price to add up, so long as you get the total money. The value range of various chickens in the topic is very clear, so a triple cycle is used to complete it.

The main implementation codes of the problem are as follows:

```
int i,j,k;//The cycle control variables of three kinds of chickens are defined to represent the number of chickens that can be bought in each cycle

for(i=1;i<=100;i++)//Enumeration cock

    for(j=1;j<=100;j++)//Enumerating hens

        for(k=1;k<=100;k++)//Enumerating chicks

            if((5*i+3*j+k/3==100)&&(i+j+k==100))//An expression that lists the total number of chickens and the total amount of money

    cout<<i<<""<<j<<""<<k<<endl;//Output reasonable combination
```

Time complexity is an important index to measure the performance of the algorithm [2]. In the competition, it is generally considered that one second is executed to the 8th power of 10. The time complexity of the above triple cycle is 1000000. If 100 is changed to N, the time complexity becomes $O(N^3)$. When n is relatively large, such as 1000, the program cannot be completed within 1 second, that is, the operation timeout occurs.

In fact, after a little analysis, we can see that the maximum number of cocks is: 100 / 5 = 20, the maximum number of hens is: 100 / 3 = 33, and the maximum number of chicks is 100. Therefore, the program can be optimized by reducing the number of cycles per cycle. The optimization of the above triple cycle can be as follows:

```
for(i=1;i<=100/5;i++)

    for(j=1;j<=100/3;j++)

      for(k=1;k<=100;k++)
```

In this way, especially when the scale n of the problem is relatively large, although the time complexity is $O(N^3)$, the reduction of the number of cycles will save a lot of time. Reducing the number of cycles is a very effective means to improve the operation efficiency in enumeration methods.

Now you must be very happy. Run quickly and see what combinations to buy chicken. Output after operation as shown in table 1:

Table 1: Output of triple cycle before verified

Cock	Hen	Chick
3	20	77
4	18	78
7	13	80
8	11	81
11	6	83
12	4	84

Carefully, you won't end here. Go back and verify it. Three chickens for one money, that means, the number of chickens can be divided by three, so 77, 80 and 83 do not meet the requirements. Enumeration requires no repetition and no leakage to get all reasonable results. Therefore, the design of conditions is very important. We need to add one in the loop: k% 3 = = 0, which correctly limits the value of chicks. The conditions in the final cycle are:

```
if((k%3==0)&&(5*i+3*j+k/3==100)&&(i+j+k==100))
```

Operation results as shown in table 2:

Table 2: output of triple cycle after verified

Cock	Hen	Chick
4	18	78
8	11	81
12	4	84

IV. USING DOUBLE LOOP OPTIMIZATION

It can be seen from the above analysis that the number of cycles per cycle cannot be optimized. We can further optimize it by using the second method in enumeration, that is, reduce the number of cycles. When the number and total number of cocks and hens are known, the number of chicks is determined. Take the number of three kinds of chickens as the enumeration object and set them as x, y and z respectively. According to the meaning of the question, the following formula can be listed:

$$x+y+z=100 \quad (1)$$
$$5x+3y+1/3z=100 \quad (2)$$

By calculating (1) and (2), z can be eliminated and x and y can be retained.

Optimization: once the number of cocks and hens i and j is determined, the chicks can only buy 100-i-j.

Key code implemented by double loop:

```
for(i=1;i<=100/5;i++)
   for(j=1;j<=100/3;j++)
        if(((100-i-j)%3==0)&&(5*i+3*j+(100-i-j)/3==100))
cout<<i<<""<<j<<""<<100-i-j<<endl;
```

Analysis: under the double cycle, the time complexity is O(N^2), and the total number of chickens (the scale of the problem) can reach 10000, which is more than 10 times higher than that of less than 1000 under the triple cycle.

IV. FURTHER OPTIMIZATION WITH SINGLE CYCLE

If the number of chickens purchased is very large, the above double cycle may also run overtime. Here, we change 100 to N, that is, n money buys n chickens and outputs the total number of combinations of chickens that can be bought, which is more universal. For the above equations (1) and (2), it is obtained after eliminating z,

$$7x+4y=n \quad (3)$$

For the deformation of (3), the expression of y can be obtained: $y = (n-7x) / 4$ (4)

Obviously, as long as you enumerate x, you can get y and further get z. As mentioned above, pay attention to the rationality test during enumeration. Since y is the number of hens, it must be an integer. Therefore, judge whether y is an integer, otherwise you may buy a hen leg.

```
for(int x=0;(n-7*x)>=0;x++)
      if((n-7*x)%4==0){//Pay attention to this detail
        int y=(n-7*x)/4;
        if((n-x-y)%3==0)//Chicks should also be rounded up
        ans++;}//Find the number of combinations that meet the conditions
```

Analysis: using single cycle, the time complexity is O(n), and n can reach the 8th power of 10.

SUMMARY

Time complexity is one of the most concerned performance indicators in programming, and it is also an important factor whether software is accepted in the market. This paper constructs the application model of enumeration algorithm through the ancient classic problem of "100 money for 100 chickens". The focus of enumeration optimization is to narrow the scope of enumeration and reduce enumeration objects.

References

[1] Lin Houcong. Olympiad in Informatics (Ke ke tong C + + version).higher education press, 1st edition, January 2018

[2] Yan Weimin, WU Weimin. data structure (C language version). Tsinghua University Press, March 2018