Analyzing the Stack Sequence Based on the Legal Stack Sequence Algorithm

Wenrong Jiang*
School of Computer and Information, Shanghai Polytechnic University, Shanghai, China
*Corresponding author

***Abstract*—The stack is a very special data structure that can only be inserted and deleted on its side, and is a very special linear table. It in accordance with the first to go out of the order of the principle of data storage, that is, first entered the data into the bottom of the stack, the last entry into the top of the stack, when the need to read data from the top of the stack to start output the corresponding data.**

***Keywords*—stack; LIFO; out stack sequence; algorithm; linear table**

I. Introduction

A stack can also be called a stack, which is a linear table with certain limits on operations. The limitation is that it only allows inserts and deletes at one end of the table. The one end of the insert and delete operation is called the top of the stack, and the other end is called the bottom of the stack. Inserting new data into a complete stack is called the entry stack, which puts the new data on top of the stack data, making this data a new round of stack top data; deleting data from a stack is called a stack or a fallback, and it removes the top data from the stack, making its adjacent data the new stack top data.

The stack is a very special data structure, it can only be inserted and deleted on its side, is a very special linear table. It in accordance with the first to go out of the order of the principle of data storage, that is, first entered the data into the bottom of the stack, the last entry into the top of the stack, when the need to read data from the top of the stack to start output the corresponding data.

Current, the form of the number of sequences needed for the stack can be solved by converting it into a tree-related counting problem, which is the number of sequence sequences that can be summed up by the sequence 1,2,...... n , that is, the former sequence is listed as 1,2,...... n the number of two forks that are not of the same type in the is very complex and difficult to understand. The algorithm for solving the stack sequence that needs to be used is obtained by means of all the complete permutations and combinations of the 1,2,...... n , and the corresponding judgment of such permutations. The time complexity of this algorithm is the same as that of solving all the complete permutation combinations, which is slower and less efficient. a new calculation method is given to improve the speed and efficiency of the work.

II. Stack

A. The Definition of the Stack

Definition: A stack is a class of linear tables that are limited to inserting data and deleting data in the head of a table. We need to figure out this idea, we must understand the "stack" the original meaning, so as to better grasp its essence. "Stack" is a place to access items or provide accommodation for tourists , the can be named as a stored warehouse, intermediate-forwarding site, so we drill it into the domain of the computer, where the data is temporarily accessible, so there is a call to stack and stack.

The contents of the data in the stack of the first system or data structure are read and inserted in two different ways. Insert data is to increase the corresponding data, this operation may only from the top of the stack, that is, the lowest address as we need to control the interface to start the corresponding operation, but read the stack of the corresponding data is arbitrary and there is no interface restrictions. More and more people will misunderstand this concept, resulting in the stack produced a lot of confusion and confusion. However, the stack in the system in the corresponding architecture of the computer has also played a leap in the role of communication between the components of the media space, that is, CPU and memory interaction hub,CPU It is possible to use a very vivid word to describe it, that is, the pipeline, the assembly line, only from the system given our own application and the corresponding entrance of the stack provided by the program to read and execute the corresponding instructions. CPU Internal communication methods need to refer to the introduction of the concepts of EU and BIU .

The stack is a very special data structure that can only be inserted and deleted on its side, and is a very special linear table. It in accordance with the first to go out of the order of the principle of data storage, that is, first entered the data into the bottom of the stack, the last entry into the top of the stack, when the need to read data from the top of the stack to start output the corresponding data. The stack also has a certain memory function, in the stack of insert data and delete data operation behavior, do not need to change the bottom of the stack pointer variable.

Stacks are very special linear tables that can be used to insert data and delete data on the same side. You can insert data and delete data operations on one side is called the top of the stack, the other side is called the stack bottom, the bottom of the stack is fixed, and the top of the stack began to change, the number of data in the stack if zero is called an empty stack. Inserting data is generally referred to as the data stack, and deleting data is called a data fallback stack or data stack. Stacks are also referred to as after entering first out of the table. Stacks can also be used to store the corresponding breakpoints when the function needs to be invoked, and to use the corresponding stack when doing recursion.

In the corresponding computer system, the stack is a dynamic memory space with the attribute mentioned above. The program that you write can press the inserted data into the stack, and you can delete the data from the top of the stack. Stack plays a very important role in the running of the program being written. The most important thing is that the stack holds some maintenance information that a function needs to be called, which is often called a stack frame or a record called an activity. The frame general situation of the stack can contain the following several aspects:

- The return address of the function and the corresponding parameter
- Temporary variables: non-static Local variables that contain functions and other temporary variables that are automatically generated by the compiler.

B. *The Nature of the Stack*

To facilitate its analysis, the problem to be studied is understood as a given sequence sequence (1,2,...,n), and the corresponding sequence of all stacks is output, that is, the stack sequence after the combination is completely arranged. When the stack size is not limited, according to the corresponding characteristics of the stack description, in fact, it is easier to get stack of the sequence needs to meet what nature.

- Nature 1: If the size of the stack is not constrained by the corresponding constraint, the sequence s1 s2 ... sn is a fully arranged combination of (1,2,......, n), then s1 s2 ... sn For the corresponding sequence of stacks, the necessary and sufficient condition is to sort the descending order of any si behind it and smaller than it.

If the size of the stack is subject to the appropriate restrictions and constraints, that is, the size of the stack m is less than the length of the sequence to enter the stack n , the sequence of the stack must first meet the nature of 1, Then, when the stack size is subject to the corresponding restrictions and constraints on the stack of the corresponding requirements of the sequence. Columns such as we need to have a length of n=5 into the stack of the sequence 1 2 3 4 5, the size of the stack m=3, when the first stack of the sequence can not exceed the elements of the data 3, That is, the first digit of the stack sequence is less than or equal to 3, that is, cannot be greater than 3, and the second must be equal to 4, not greater than 4. Normally, the first digit is less than or equal to m , the second is less than or equal to m+1, and so on, until the sequence of the stack is n-m bit is less than or equal to n-1 The time.

- Nature 2: If the size of the stack is subject to the corresponding restrictions and constraints, that is, the size of the stack m is smaller than the length of its stack sequence n , the sequence s1 s2 ... sn is (1,2,......, n), the s1 s2 ... sn is called a sufficient and necessary condition for the stack sequence to satisfy the nature 1 , and the first J bit of the sequence is less than or equal to m+ J-1.

C. *A Valid Stack Sequence Algorithm*

Our premise is to enter the stack must be in a certain order into the stack, columns such as 1,2,3,4,5 in the order to enter the stack, If the first stack of data is 5 , we necessarily need the sequence of advanced stacks to be 1,2,3,4,5, Then this sequence is on the stack, so that the first stack of data is 5, no other data can be entered into the stack, so the stack needs to be out in order, so the order of the stack is 5,4, 3,2,1 .

If the order of the stacks is 4,5,3,2,1 , then we need to analyze the first one out of 4 . We have to first enter the stack for 1,2,3,4 , and then stack the 4 . Then let 5 go into the stack, and then press 5,3,2,1 in the order of the stack.

Also if the order of the stack is 4,3,5,2,1 or 4,3,2 ,1,5, you also need to first enter the 1,2,3,4 order into the stack, and then 4,3 output Stack, if this is the previous stack sequence, let 5 enter the stack, and then 5,2,1 output Stack, if the subsequent stack sequence will be 2,1 output stack, and then let 5 Enter the stack, then the 5 output Stack, This will give us the stack sequence we envisioned above.

We can use the following methods to determine:

If the order into the stack is 12345, given a sequence of stacks, such as 35421, it is a legitimate sequence. Because for any number of sequences in the stack data, all the numbers behind it, which are smaller than it, must be in descending order.

Like what:

If the sequence of stacks is 51234, obviously it does not meet the requirements given above, because for 5 , the sort sequence of the number smaller than the5 is a sequential sequence of 1234. So it's not a legitimate stack sequence.

If the sequence of the stack is 41532, it is also an illegal stack sequence, because 4 is followed by a smaller 1,3,2 than a descending sort sequence.

This approach is a very simple and effective way to eliminate the wrong stack sequences.

If the input sequence of the stack is 1,2,3,...,n, the design algorithm will find all possible stack sequences, that is, the entire stack sequence.

Assuming n = 5, the stack sequence may have 1 2 3 4 5, 1 2 3 5 4, 1 5 4 3 2, and 1 4 5 3 2 and so on

If you use a recursive method to do this, the main idea is: from 1 to n input, each number may only correspond to two operations, one is into the stack operation, the other is the operation of data stack, we use a stack to save the data into the stack, An array stores data from the stack.

D. Solution for Stack Sequence Problem In data Structure

Solving the problem of stack sequence with elimination method

We can also use the combinatorial knowledge in mathematics to solve this problem, in the real use is not simply need this kind of knowledge, but also need to engage a lot of simple mathematical problem solving skills, such as exclusion.

In the problem of stack sequence, when the value of n is large, there is a very large number of combination results, we can first list some of the relatively simple sequence combinations, and then use a certain calculation formula to calculate the n data may appear when the combination of series. But the combination of these series is not necessarily meet our requirements, the need to meet the stack of advanced to go out after the principle of the line. Specific analysis can be done through a one by one arrangement:

When n=1 , there is only one sequence of permutations and combinations, that is, 1, and the sequence number of the stack is only one; when n=3 , there is a different arrangement, that is, 123, 213,231,321,132 ,312, in which 312 The combination of this way is not in line with the stack of the first to go out of the principle, it can be excluded, so the actual stack number of the sequence has 5 different order, then the number of the stack sequence is less than the number of permutations and combinations.

After enumerating through the above sequence, we can draw a simpler two conclusions: (1) The number of stack sequences is less than or equal to the total number of stack sequence combinations; (2) when a larger array is encountered, the entire sequence is usually rendered in descending order.

III. Solution Analysis

A. Topic Description

Poj1363 This topic is simulated into the stack and out of the stack of the process, will be 1, 2, ..., N into the stack, the results of a stack of many kinds, input a set of data sequences to determine whether this set of data sequence to meet the results of a group of stacks.

B. The Idea of Solving Problems

The station platform as a corresponding stack, in the order of 1-n to traverse the original order of the train, that is, the order into the stack. If the number of the train on the top of the stack and the number of outbound numbers we give are ordered. We compare the number of the departing station with the number we give, when the train is out of the stack, and then all the trains that meet the outbound numbers are left on the stack, otherwise we will always be in the stack state. The final check of the input train number order is not all out of the stack. If the stack is out, the output is Yes, otherwise the output is No.

C. Code Analysis

- Establish two arrays x,y, by topic given n<1000 you can define the size of an array to be 1005 , a train number sequence to stack , and another stores the train number sequence of stacks.
- reads the value of n , and the first for loop stores the number sequence of the stack train in the y Array, preparing for a subsequent test to satisfy one of the stack sequences. The second for loop controls the train number sequence to check for conditions and enters The while loop when I controls the size of the number and satisfies x[m]==y[j] , where while the loop controls the flow of data entered, the first if statement, when m>0 , controls m minus reduction to control X[m] Changes in the movement and size of the array in the loop, and when m<=0 is assigned to M[x] , the assignment is 0, and on m minus ,the makes it normal to jump out of the loop when m = 0 , without the state of the dead loop, and then J Gaga is to control the movement and size of the data in the Y[j] in the loop, and the second if statement, when M==-1 executes an action that jumps out of the while Loop , and the goes to the second for loop to continue executing the program, Until the For loop executes, jumps out, enters the last if statement, and determines the size of the J and N if J and N such as large description of the input train number sequence satisfies the condition, is one of the stack sequence, output Yes, otherwise output No.
- the size of the for x,y array range, because the topic is given a n<1000, so it is defined directly as an integer array and the range size is positioned 1005, many 5 is intended to prevent accidental occurrences.

X[m],y[j] control the input data against the data that can be derived from the stack, and continue to run if the condition is met, otherwise the value of the last for loop until the end,X[m] is continuously valued by I , which can improve productivity, Make the program more simple and clear.

D. Diagram and TABLE

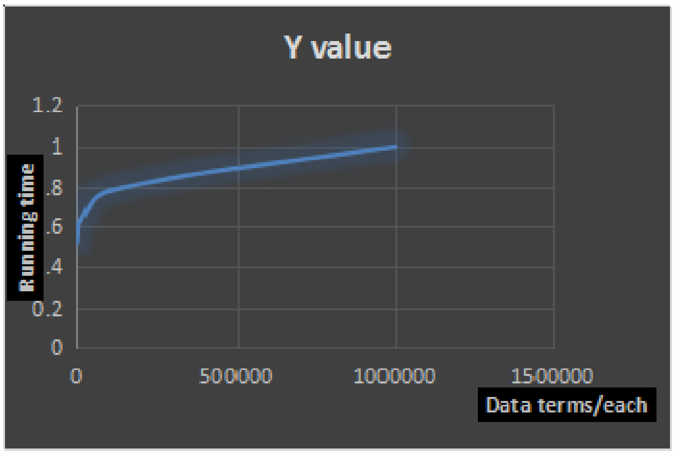

FIGURE I. RUNNING TIME

TABLE I. DATA TERMS & RUNNING TIME

Data terms	100	1000	10000	100000	1000000
Running time	0.52	0.55	0.63	0.78	1.00

It is clear from the chart that, as data items continue to grow, the time spent on computers is growing.

E. Code Fragments

```
if (flag)// The premise is reasonable to judge, if it has been unreasonable read-only such as data, do not judge
{
if (stack_len!= 0 && stack[stack_len-1] = tmp)// if the stack is not empty and the next stack number is the same as the read number, the stack
  {
        --stack_len;
        last = tmp;
        max>?=last;
   }
else if (tmp > last)// If the number read is larger than the maximum number of the stack, the number before the number is read into the stack, which is the number as a stack number
  {
        for (int i = max+1 i < tmp; ++i) {
            stack[stack_len++] = i;
        }
        max = last = tmp;
  }
else// If none of the above two conditions are met, this sequence does not satisfy the requirement
{
       flag = 0;
}
```

IV. CONSIDERATIONS

- Note which variables require global variable naming, and which variables only require local variable naming;
- When data is defined, some variables must be initialized for definition, or the program may complain.
- Looping statements are best enclosed in parentheses to avoid unnecessary hassles by having more or less parentheses.
- Try to set variable names with simple word letters.

V. SUMMARY

Stack is a abstract data type, abstract data type and data structure based on the principle of last in first out(LIFO). Considered as a linear data structure, or more abstractly a sequential collection, the push and pop operations occur only at one end of the structure, referred to as the top of the stack. This makes it possible to implement a stack as a singly linked list and a pointer to the top element. A stack may be implemented to have a bounded capacity. If the stack is full and does not contain enough space to accept an entity to be pushed, the stack is then considered to be in an overflow state. The pop operation removes an item from the top of the stack. Stacks are widely used in real life, such as bullet clips, process management of operating systems, and so on.

ACKNOWLEDGMENT

This work is supported by the Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University under Grant No. XXKZD1604.

REFERENCES

[1] Wenrong Jiang, Jian Chen, Wei Tao. The Development of Automotive Interior Sales Website[J]. Communications in Computer and Information Science (Volumes 268), 2011(12):342-348. EI:20122015019844

[2] Wenrong Jiang. App Tamper Detection and Retrospective Program Research[C]. International Conference on Computational Data and Social Networks (CSoNet 2018). Springer. December 18, 2018, 318-329.

[3] Wenrong Jiang , Wenting Shao. Analysis and Evaluation of Development Degree of the Special Township in a Prefecture-level City[J]. 2018 2nd International Conference on Applied Mathematics, Modelling and Statistics Application (AMMSA2018), Sanya, China, ATLANTIS Press. May 28, 2018, 265-270.

[4] Wenrong Jiang, Shijie He. The research of small and medium-sized wedding photography enterprise's product types and consumer participation behavior[C]. Proceedings of the 2016 International Conference on Social Science, Humanities and Modern Education(SSHME 2016), Guangzhou, China. Atlantis Press. May 30, 2016, 103-107.WOS:000390400800020.

[5] Wenrong Jiang. App Tamper Detection and Retrospective Program Research[C]. International Conference on Computational Data and Social Networks (CSoNet 2018). Springer. December 18, 2018, 318-329