DOI: 10.53469/wjimt.2026.09(02).01

On the Study of Data Structure

Long Wen

Jincheng College, School of Computer and Software, Sichuan University, Chengdu 611731, Sichuan, China

Abstract: ***Data structure is a set of data elements which have one or more relationships with each other and the relationship between data elements in the set. Data structure is to teach you how to write a better algorithm, and a good procedure is the algorithm + data structure reflected, so the study of data structure is extremely important, the data structure of the classification is particularly much, extremely easy to confuse.***

Keywords: Linear structure; Nonlinear structure; Single-chain tables; Trees.

1. INTRODUCTION

Nowadays, the Internet era is developing rapidly, and the job of a programmer is especially important in the 21st century. And there is a lot of potential, a lot of demand in the market, and the development of science and technology, the progress of the program, and so on, and the programmer the most important is to learn to program, and a good program is the need for data structure + algorithm together reflected, And a procedure is good or bad is to provide some conditions in the data structure, and to modify, so that the procedure becomes better, faster, better to operate, then the data structure is so important that everyone can learn? Is it accessible to everyone? So how can we make it easier to master learning methods?

2. COMPOSITION OF DATA STRUCTURES

2.1 Concepts in question

(1) Data: Any symbol that can be input into a computer and processed by a computer program.

(2) Data element: The basic unit of data is called a record or a node.

Data structure: A collection of data elements that have one or more specific relationships with each other.

2.2 Composition of data structures

Data structure is divided into three parts: logical structure, physical structure, calculation.

2.2.1 Logical structure

It exists independently of computers, and its meaning is to describe some sort of logical relationship between data elements, usually composed of a binary group, such as:

A=(D,R), D={A,B,C,D,E,F,H}, R={(D,A),(D,G),(D,A),(A,C),(G,E),(G,H),(E,F)}

2.2.2 Physical structure

Physical structure It requires a computer, and it refers to how the logic of the data is represented in computer storage.

Specific methods are: sequence method, link method, hash method (hash method)

The following are two commonly used methods: sequential and linking The sequential method refers to data elements that are logically adjacent and physically adjacent. The advantage of this method is that it can be accessed at random, but it is difficult to insert and delete.

The link method is to divide each unit of the structure into two, one part of the pointer field points to the address of the next element, the other part stores the data information of the element。[1]。

2.2.3 Algorithms and Evaluation

Algorithm: refers to a finite sequence of operations (the idea of a method for solving a problem) that solves a particular type of problem.

Difference between algorithms and programs: Algorithms need to be processed before they can become programs, while programs can be run directly, but algorithms cannot, and algorithms are often described as functions.

The algorithms are evaluated on the basis of: readability, robustness (effective judgment of the content entered by the user), correctness, time complexity and spatial complexity (effectiveness).

A type of arithmetic that is in the order of magnitude of the number of times a base statement is rerun, usually expressed as T (n) = O {f (n)}.

For: (i = 1;i<=n;i++) X++; F (n) = n, T (n) = O (n). This is the linear order.

For: (i = 1;i<=n;i++) for (j=1;j<=n;j++) X++; f(n)=n*n, T(n)=O (n*n).

3. LINEAR AND NONLINEAR STRUCTURES (EXAMPLES)

3.1 Linear Table

(a1, a2, a3……ai……an)

(1) The meaning of a linear list: a finite sequence of n data elements of the same nature formed.

(2) Representation of linear table:

"n" represents the length of the linear table, when n = 0, the table is empty.

"i" represents the position of the data element in the linear table.

(3) Linear representation of the relationship between elements: next to each other.

3.2 The physical structure of a linear table

(1) A linear list stored sequentially is called a (sequential list) and it requires an array.

(2) Advantages and disadvantages: 1. Random access, fast, 2. Insert and delete are not convenient (such as moving a large number of elements).

3.3 Implementation of the algorithm

(a1, a2, a3, ak-1, ak, ak+1,an)

Assignment method:

let a1 = 9, a2 = 2, a3 = 6, a4 = 5, a5 = 7, let K = 3
It becomes a1 = 9, a2 = 2, a3 = 5, a4 = 7

Code:

Delete（int a[],int n, int k） Int i;If(k<1||k>n)Printf ("Delete failed")//The element to be deleted is not within this range else{for(i=k;i<n;i++) a[i-1]=a[i];}n--;

3.4 Trees and Binary Trees

The definition of a tree: a finite set of n nodes. There is a single node called the root.

3.5 Concepts related to trees

Cutting points: roots, children, parents, grandchildren, brothers, leaves, etc. The degree of a node: The number of children of the node is called the degree.

Degree: The maximum value of the degree in the tree

The hierarchy of the nodes: the root is the first layer, the children of the root are the second layer, and so on.

Root depth: The maximum value of the node hierarchy in the tree.

3.6 Binary tree (maximum degree is no more than 2)

Definition: A tree with a maximum of two subtrees per node in a tree.

Features: 1 The maximum degree of each node does not exceed 2,2 Son Trees: There are left and right branches, and the order cannot be reversed.

4. LEARNING METHODS

4.1 Understanding knowledge is the basis, brushing is the auxiliary

To learn data structure well, first of all, to put the basics of the whole solid, the basics are the most important, from reading so many years, there is no subject is not to learn the basics can be learned well, and when you learn the basics, When brushing questions, you can improve your efficiency and correctness. When you start learning, be sure to understand the criteria for judging the goodness of a program, such as time complexity and spatial complexity, which are the criteria to judge the goodness or badness of another program. You can brush more knowledge of this aspect during the early learning.

When learning about linear structures, such as linear tables, there are several steps that need to be taken:

(1) Drawing

(2) Split a large operation into several small operations

(3) Decide clearly what methods, tools and tools are used for each small part

(4) Frameworks for writing code

(5) Verify

Among them, drawing is used to help you understand the content of the problem, and dividing a large operation into several small operations can be better done step by step. Write the framework faster, and the most important step is the verification of the last step. This is a step that many students don't take after writing it. This step is extremely important. There is a chance that a small condition will cause the entire algorithm to fail, and the programming will not be implemented at the end.

Master each type of expression and ideas The types of large data structures are divided into linear and nonlinear structures, In the case of linear structure, I gave an example of a linear table. When learning to insert a linear Table, insert in a lineartable. First, consider which of the two types of tables is easier to implement, and then we think of chain storage, chain storage methods, We need to use structures and pointers, so we need to have knowledge of both. When we build a chained stored table, we can insert either positionally or sequentially. In the case of an ordered insert, we need to reference two pointer variables.

The code is as follows:

Insert (NODE *p, int x) {NODE *P=head, *S; While (p-> next & & p-> next-> data < x) p=p->next; S= (NODE*) malloc (sizeof (NODE)); >data=x; >next=p->next; P->next=s;}

5. CONCLUSION

That's what I learned about data structures. Data structures are extremely important to programmers. It is not just an algorithm, it also reflects the "soul" of a procedure, how a procedure is compiled, it will be fully witnessed, so the study of data structure requires constant summary, brush questions, get their own things.

REFERENCES

[1] Weimin Yan, Dongmei Li, Weimin Wu. Data Structure (c language version | second edition).