On Radix and Straight Insertion Sort Programming Based on Linked List and Queue Technique

Wei Canmei
Guilin University of Technology
Vocational College of Technology
Guangxi， Nanning
434402695@qq.com

Yang Yahui
Guilin University of Technology
Vocational College of Technology
Guangxi， Nanning
yyh@gliten.com

***Abstract*—Linked list technique and queue technique are two tool techniques of used to assist realization of many algorithms. Sort algorithm is the premise of a lot of application software meeting function and performance requirements. This paper discusses fused application of the two techniques and knowledge to make the characteristics of these techniques and knowledge more outstanding and to make the effect more powerful so as to provide efficient support tools and techniques for improving operating efficiency of application software.**

***Key words*—Component, linked list technique, queue technique, radix sorting, straight insertion sorting, tool**

I. INTRODUCTION TO THE PROBLEM

Among major techniques of data structure, linked list technique, queue technique and sort algorithm are relatively important and must be mastered. Besides, they have close relation: linked list technique and queue technique are support tools of sort algorithm, while sort algorithm verifies one of main applications of linked list technique and queue technique in turn – tool.

However, in general literatures, these techniques and algorithms are arranged and introduced separately. Thus, their inherent lose relation cannot be revealed. Especially during introducing linked list technique and queue technique, the application examples cannot accurately reveal tool essence of linked list technique and queue technique. Therefore, when learning and grasping these techniques, it is often hard for beginners to understand practical application of linked list technique and queue technique. In addition, they have to specially spend time in learning and grasping sort algorithm.

II. SOLUTION TO THE PROBLEM

Aiming at the above problem, through researches and exploration, we optimize and integrate linked list technique, queue technique, radix sorting algorithm and straight insertion sorting to arrange and introduce them together. Besides, we also design corresponding examples to fuse these techniques. In this way, linked list technique, queue technique and part of sort algorithm are introduced meanwhile. This can shorten the time and reflect the major effect of linked list technique and queue technique – tool of realization of algorithms.

Firstly, the core contents of linked list technique, queue technique, radix sorting algorithm and straight insertion sorting are presented.

III. EASE OF USE

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

A. Linked list technique

Prior to introducing linked list technique, sequential storage technology must be introduced. Sequential storage technology and linked storage technology are both storage ways of data in internal memory. Sequential storage technology means the data in the same application is stored in a memory block with continuous address. Besides, this is generally gained through adopting a static application way. In other words, it is necessary to first estimate the data size to be stored to apply for appropriate memory space. One of the shortcomings of sequential storage technology lies in the lack of flexibility. The size of memory block applied for is fixed. If the data size in the application increases, the situation of insufficient storage space may occur (called overflow phenomenon); conversely, if the data size decreases, the waste of internal memory space may be caused. The second shortcoming of sequential storage technology is low time efficiency. When it is necessary to insert or delete data in and from the sequence list, it is required to first shift part of data ahead or backward and then the data can be inserted or deleted. The operating speed of the program is slow.

Linked storage technology is also a storage way of data in internal memory, but the internal memory space needed by such storage way is not gained through static application in advance. The application is implemented only when the data need to storing. The data in the same application do not need consecutive storage, but are dispersedly stored at different

locations in the computer memory. Since the data in the same application may be dispersedly stored in available space in internal memory. To reflect the data in the same application, extra internal memory space needs applying for to store the memory address of the next data. Data storage space in the same application is linked one by one. Such link structure is called linked list vividly. Linked storage technology will waste some internal memory space, but it owns the following advantages:

- Firstly, flexibility. In linked list (Linked storage) technology, the internal memory space is gained through dynamic application when the data are about to be stored in the internal memory. As long as there is unoccupied space in computer memory, the necessary space can be gained through application. For a general application, computer memory is enough. The overflow phenomenon rarely occurs. If the data in the linked list is deleted, the internal memory space the data occupy will release timely and will not occupy all the time. This improves the use ratio of computer memory greatly. Thus, the data size in the same application may be large or small, and the phenomenon of large scale of waste will not appear.
- Secondly, efficiency. When inserting or deleting data in or from the linked list, it is only necessary to modify straightional address of relevant several pieces of internal memory space, without need of large scale of data shift. This saves operating time greatly.
- Thirdly, diversity. Through different straightional forms, single linked list, double linked list and orthogonal list can be formed to meet storage forms of different applied data, thus providing premise technological base for expanding application area of computer software.

B. Queue technique

Queue technique aims to queue the data in the internal memory. The data first stored rank the head of the queue and the later data rank the tail of the queue. When the data are deleted, the data at optional position in the queue cannot be deleted at will. The data at the head of the queue must be deleted first (i.e. first in first out). Queue technique is mainly used to assist realization of some algorithms, belonging to tool technique. Data queue can be implemented in the sequence list or in the linked list. No matter which way is adopted, there are specific advantages and disadvantages. Sequential queue is simple but lacks flexibility, with low time efficiency; linked queue is flexible, and the length is not limited; and program run has high time efficiency.

C. Radix sorting algorithm

Radix sorting is an internal sorting method to sort single logic key value based on multiple key value sorting. It is realizes with the help of "allocation" and "collection".

Thought of radix sorting algorithm: the key value can be regarded as being compounded by several key values. For example, if the key value is a numerical value between 0 and 999, K can be regarded as being composed of three key values (K1, K2, K3), where K1 is a hundreds digit; K2 is a tens digit; K3 is a units digit. The range of each key value is the same ($0 \leq K1, K2, K3 \leq 9$).

MSD method and LSD method can be used to realize radix sorting algorithm. Sort according to LSD method: from the key value of the least significant digit, "allocate" each record in the sequence into d queues according to the key values and then "collect" them to the queue; repeat it d times and sort the key values according to a sequence.

For radix sorting algorithm in general literatures, the sorting is conducted by taking the positive integer with less than three digits for example. Thus, the coverage is insufficient, lacking certain representativeness.

D. Straight insertion sorting algorithm

Basic thought of straight insertion sorting algorithm: insert the record to be sorted into an ordered sequence according to the size of the key until all records are sorted well. The main operation for realization of insertion sorting algorithm is to compare and insert. If the sequence list technology is used, certain data shift will certainly be carried out, thus resulting in low efficiency of algorithm. If linked list technique is adopted, data shift is unnecessary during insertion, thus reflecting the major advantage of linked list technique and also standing out tool essence of linked list technique.

Secondly, we fuse these techniques to design the program, fully utilize and give full play to advantages of various techniques and algorithms and realize maximization of these advantages and significant practical application.

The following is the program we design which is characterized by:

1) Apply the basic techniques related to the linked list: initialization of linked list, establishment of initial single linked list, insertion and deletion of nodes of single linked list etc.

2) *Apply the basic techniques related to linked queue:* initialization of linked queue, en-queuing and de-queuing of linked queue etc.

3) Radix sorting algorithm is realized with the aid of single linked list technique and linked queue technology. The first improvement of this algorithm is to store raw data with single linked list. Any data can be sorted. This method solves the defect of data overflow caused by the use of sequence list. The second improvement is to sort multi-digit positive integer, solving the limitation that general radix sorting algorithm can only sort the positive integer with less than three digits.

4) Straight insertion sorting algorithm is realized by the aid of linked list technique. Compared with realization the algorithm with sequence list, this method can reduce the time for data shift during data insertion and also solves the defect of data overflow caused by the use of sequence list.

5) *The two sorting techniques are used at the same time.* The menu can be provided to users for their options. The introduction and realization can be conducted through analogy so as to save the time of learning and comprehension for the beginners.

Source code as shown below:

```
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#include "windows.h"
typedef  int  Elemtype;
 /*The description of the nodal structure of single linked
    lists*/
typedef struct node
{  Elemtype  data;
   struct node  *next;
 } LinkNODE;
   /*The description of the linked queue on the fist node
    structure*/
 …
   // Initialize the linked queue
void init_Q(LinkQ **f)
{  int k;
   LinkNODE *v;
   for(k=0;k<10;k++)
    { v=(LinkNODE *)malloc(sizeof(LinkNODE));
      v->next=NULL;
      f[k]->front=f[k]->rear=v;  }
 }
   // Initialize the single linked list
LinkNODE *init_L( )
{ … }
   // Output the data of the single linked list nodes
void out_L(LinkNODE *head)
{  … }
 /* Establish the single linked list, and the same group of
    data will be established as two independent single
    linked lists in internal memory */
int create_L(LinkNODE *head,LinkNODE *head2)
{  int  x,max,d=0;
   LinkNODE *v,*u;
   scanf("%d",&x);
   max=x;
   while(x!=-1)
   {   // Establish the first single linked list
       v=(LinkNODE *)malloc(sizeof(LinkNODE));
       v->data=x;
       v->next=head->next;
       head->next=v;
       if(max<x)   max=x;
```

```
 // Establish the second single linked list
       …
       }
    /* Obtain the bit of the maximum positive integer and
     prepare for the radix sorting */
     …
    return d;
}
  // The algorithm for isolating a certain bit of digit
int num( int m, int i )
{ int n,s=1;
 //Isolate the required data bit (bit i) through circulation
   while(s<i)
   { m=m/10;   s++;  }
   n=m%10;
   return(n);
 }
 // The data allocation algorithm for radix sort
void fenpei(LinkQ **f,LinkNODE *head,int i)
{  LinkNODE *v;
   int  n;
    v=head->next;
    /* The data of the circulation pair single linked lists shall
       be allotted into the corresponding linked queue in
       accordance with corresponding bits */
     while(v!=NULL)
      {  n=num(v->data, i);
         f[n]->rear->next=v;
         f[n]->rear=v;
         v=v->next;
         f[n]->rear->next=NULL;  }
 }
   // The data collection algorithm for radix sort
void shouji(LinkQ **f,LinkNODE *head)
{  int k,j;
   k=0;
   //Search for the first nonblank linked queue by judging
   while(f[k]->front->next==NULL&&k<10)      k++;
  // Collect the nodes of the first nonblank  linked queue
    …
   /* Collect the nodes of the other nonblank linked queue in
    a circulation way */
    …
}
// Straight insertion sorting algorithm
```

```
 void isort(LinkNODE *head2)
{  LinkNODE *p,*w,*temp;
   p=head2->next;
    /* Judge whether the previous data is in order or not. If
       the data is in order, then skip them one by one */
   while(p->next!=NULL && p->data>=p->next->data)
    {   p=p->next ; }
     /* The operation of the comparison and insert shall
       be used for the later unordered data, and these data
       shall be inserted into the previous ordered data
       sequence */
   if(p->next!=NULL)
    {  w=p->next;
       p->next=NULL;
       while(w!=NULL)
       { temp=w->next;
          p=head2;
          // Search for the correct insert position
         while((p->next!=NULL)&&(p->next->data>w->data))
             p=p->next;
         w->next=p->next;
         p->next=w;
         w=temp;   }
      }
}
void main( )
{  LinkQ  t[10],*e[10];
   int k,i,d,s1,s2;
   char yn='y';
   LinkNODE  *head1=NULL,  *head2=NULL;
   // Initialize
   ...
 while(yn=='y'||yn=='Y')
{ system("CLS");
  head1=init_L();    head2=init_L();
  printf("***...****\n");
 printf("* 1.Radix sorting algorithm  2.straight i nsertion
    sorting algorithm  3.Exit  *\n");
 printf("***...****\n");
 printf("Please input the operation serial number you
    selected :");
 scanf("%d",&s1);
 if(s1==1||s1==2)
   {  printf("  Please input ...\n  ");
      d=create_L(head1,head2);  }
  switch(s1)
    {  case 1:
       printf("\n   *** The raw data is ***\n    ");
          out_L(head1);
          for(i=1;i<=d;i++)     // Radix sorting
           {  fenpei(e,head1,i);
              shouji(e,head1);
               init_Q(e);   }
printf("   *** Use the radix …***\n    ");
    out_L(head1);
    getch();
    printf("***...**** \n");
    printf("  *  1. Continue straight insertion sorting algorithm
       2. Return *\n");
    printf("***...**** \n");
    printf("  Please input … you selected :");
    scanf("%d",&s2);
    if(s2==1)
    {   isort(head2);   // Straight insertion sorting
        printf(" \n** Use the insert sort… **\n     ");
        out_L(head2);
         }
      getch();
      break;
    // Straight insertion sorting
  case 2:
      …
  default :yn='n';
     }
   }
}
```

Program has been in the integration of editing environment Visual C++ 6 compiled to run through. Below is a screenshot program testing process section.

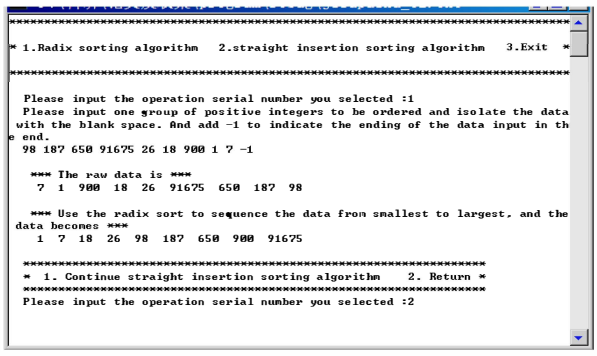

Figure 1. The radix sort first and then the straight insert sort are shown

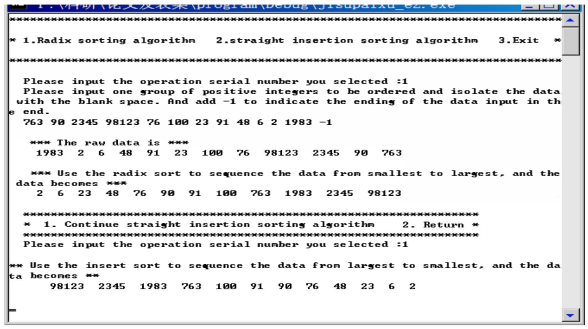

Figure 2. Only the radix sort without going on with straight insert sort are shown

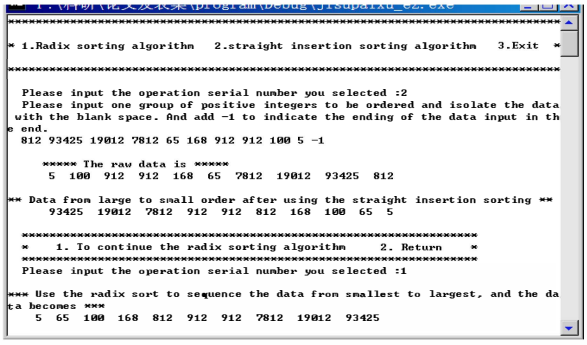

Figure 3. The straight insert sort and then the radix sort is shown

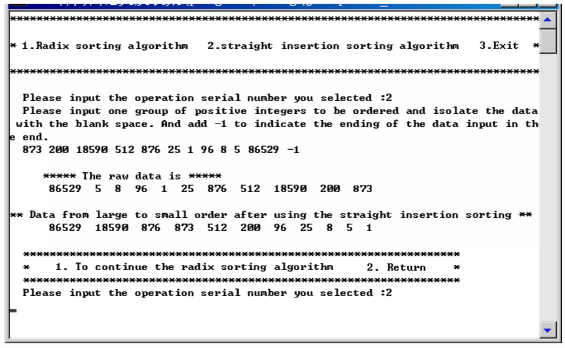

Figure 4. Only the straight insert sort without going on with radix sort are shown

Figure 5. No sort operation is shown in figure 5. May chose to exist straightly, increasing the user's selectivity

IV. CONCLUSION

The general literatures do not introduce the linked list technology, queue technology and sort algorithm together. This paper will discuss the feasibility that introduction the three together according to this background. The main viewpoints shall be concluded as follows:

The linked list technology is a kind of data storage on the computer internal storage, which makes the computer's memory space be used flexibly and improves the management efficiency of memory space.

The queue technology and sort algorithm are the tool and mean for achieving the algorithm and operation. With the help of them, it will make the computer achieve the functions that required by people.

This paper discusses the combination of three, which will provide supporting technology of integration for many application software, so as facilitate the development efficiency and operational efficiency of the application software.

The main shortage of this paper:

1) The algorithm in this program can be improved so as to increase the operational efficiency of program.

2) The analysis on the time and space efficiency of radix sort algorithm and straight insert sort algorithm in program has not been conducted. From the perspective of program design, lack some completion and contentment. The supplement will be supplied in the future.

REFERENCES

[1] Adam Drozdek, Data structure and algorithm - C++ edition (Edition 2), Beijing: Tsinghua University Press, 2003

[2] Yan Weimin, Wu Weimin, Data structure – C language edition, Beijing: Tsinghua University Press, 2011.

[3] Xu Cuixia, Data structure case course, Beijing: Peking University Press, 2009.

[4] Yang Zhenghong, Data Structure, Beijing: China Railway Publishing House, 2002.

[5] Ren Xueping, Wang Libo, Zhao Baohua, educational reform of Data Structure" course with PIC-CDIO idea [J], Computer Education, 2012 (12), 29-32.