DS HW12

Deadline: 2019/1/1

手寫題

2. An array contains the elements shown below. Show the contents of the array after it has gone through a one-increment pass of the shell sort. The increment factor is $k = 3$.

```
23 3 7 13 89 7 66 2 6 44 18 90 98 57
```

4. An array contains the elements shown below. What would be the value of the elements in the array after three passes of the heap sort algorithm?

```
44 78 22 7 98 56 34 2 38 35 45
```

6. An array contains the elements shown below. Using a quick sort, show the contents of the array after the first pivot has been placed in its correct location. Identify the three sublists that exist at that point.

```
44 78 22 7 98 56 34 2 38 35 45
```

8. After two passes of a sorting algorithm, the following array:

```
80 72 66 44 21 33
```

has been rearranged as shown below.

```
21 33 80 72 66 44
```

Which sorting algorithm is being used (straight selection, bubble, or straight insertion)? Defend your answer.

10. Show the result after each merge phase when merging the following two files:

```
6 12 19 23 34 · 8 11 17 20 25 · 9 10 15 25 35
```

```
13 21 27 28 29 · 7 30 36 37 39
```

12. Rework Exercise 11 using the balanced merge method.

程式題

19. Repeat Problem 14 using the quick sort (see Program 12-8).

25. Merge sorting is an example of a divide-and-conquer paradigm. In our discussions, we used merge only as an external sorting method. It can also be used for internal sorting. Let's see how it works. If we have a list of only two elements, we can simply divide the list into two halves and then merge them. In other words, the merge sort is totally dependent on two processes, distribution and merge. This elementary process is shown in Figure 12-24.

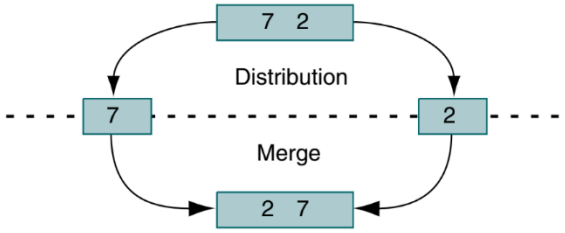

FIGURE 12-24 Split and Merge for Project 25

Given a list longer than two elements, we can sort by repeating the distribution and merge processes. Because we don't know how many elements are in the input list when we begin, we can distribute the list originally by writing the first element to one output list, the second element to a second output list, and then continue writing alternatively to the first list and then the second list until the input list has been divided into two output lists. The output lists can then be sorted using a balanced two-way merge. This process is shown in Figure 12-25. It could be called the "sortless sort" because it sorts without ever using a sort phase.

Write a C program to sort an array of 500 random numbers, using this approach. Print the data before and after the sort.

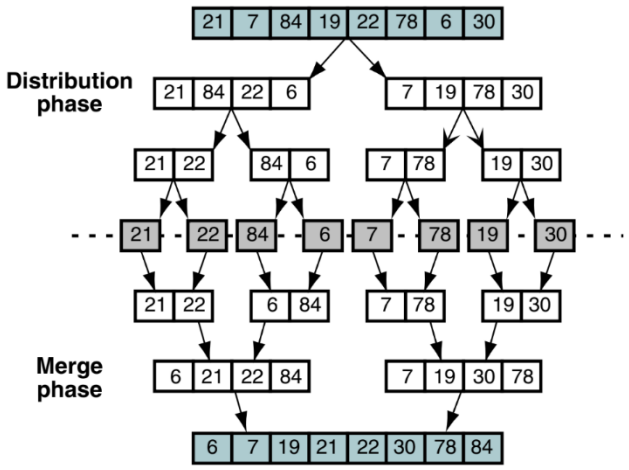

FIGURE 12-25 Sortless Sort

参考

11. Starting with the following file, show the contents of all of the files created using external sorting and the natural merge method (do not include a sort phase):

```
37 9 23 56 4 5 12 45 78 22 33 44 14 17 57 11 35 46 59
```

14. Modify Program 12-3, "Insertion Sort," to count the number of data moves needed to order an array of 1000 random numbers. A data move is a movement of an element of data from one position in the array to another, to a hold area, or from a hold area back to the array. Display the array before and after the sort. At the end of the program, display the total moves needed to sort the array.