Data structure
Workshop 8

I. Exercise

1.1 Using the key sequence (503,087,512,061,908,170,897,275,653,426) as an example, manually execute the following sorting algorithm and record the key state at the end of each pass:

(1) Direct insertion sort;
(2) Shell sort (increment d[1]=5);
(3) Quick sort;
(4) Merge sort;
(5) Heap sort;

1.2 Among the sorting methods listed in question 1.1, which are stable and which are unstable? For each unstable method, provide an example of its instability.

1.3 Clearly, the number of comparisons required for quicksort on a sequence of n elements depends on their initial arrangement.

(1) Under optimal conditions, how many comparisons are required when n=7? Please provide the rationale.
(2)Give a best initial arrangement instance for n=7.

1.4 Question: Based on the tournament ranking principle, how many matches must be scheduled to determine the final standings among eight athletes, considering the worst-case scenario?

1.5 Determine whether the following sequence is a heap (small top heap or large top heap). If not, adjust it to a heap (with minimal record exchange).

(1)(100,86,48,73,35,39,42,57,66,21);
(2)(12,70,33,65,24,56,48,92,86,33);
(3)(103,97,56,38,66,23,42,12,30,52,06,20);
(4)(05,56,20,23,40,38,29,61,35,76,28,100).

II. Experiment

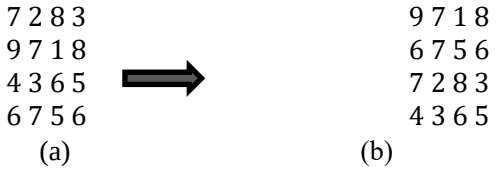
2.1 Implement the simple selection sort algorithm by using single linked list as storage structure.

2.2 Consider a 2D array defined as int A[n][n] (where n <500). Design a sorting algorithm called

ArraySort (int A[500][500], int n)

that follows these rules:

First, calculate the sum of all numbers in each row, then apply quicksort to sort the rows in non-increasing order. An example is shown in the figure below, where (a) represents the original array and (b) displays the sorted result after applying the sorting rule.



(a) (b)