台灣聯合大學系統 110 學年度碩士班招生考試試題

類組：電機類　科目：資料結構(3002)　**問答題**　共 2 頁 第 1 頁

1. A string matching algorithm aims to find the occurrence of a string P within another string S.
   (5%) (a) Use pseudo-code or C code to describe a string matching algorithm that has time complexity of O(N_P*N_S), where N_P and N_S are the lengths of P and S, respectively.

   (5%) (b) The Knuch-Morris-Pratt (KMP) algorithm is a string matching algorithm designed to reduce redundant symbol comparisons in string matching. What is the big-O time complexity, in terms of N_P and N_S, of the KMP algorithm?

2. (10%) The C structures of a singly linked list and its list items are given below:
```
struct IntListItem {
  int value;
  struct IntListItem * link;
};
struct IntList {
  struct IntListItem *front;
};
```
   Give the C implementation of a function
   `void smallest_to_front(IntList *list)`
   that moves the item with the smallest `value` to the front of the list, and leave the relative orders of the other list items unchanged.

3. (10%) Given a non-empty queue Q and a stack T, use pseudo-code to describe an algorithm to reverse the order of the elements in Q using T. For both Q and T, you can only use their push and pop operations.

4. The available-space lists for linked lists are used to "recycle" unused list items instead of releasing their memory.
   (3%) (a) Describe the advantage of using available-space lists.

   (3%) (b) Describe in pseudo-code of how to implement the function `clear`, which should run in O(1) time complexity. This function clears the content of a list so that its size becomes zero.

   (4%) (c) Describe in pseudo-code of how to implement the function `add_space(N)`, where `N` is a positive integer. This function adds `N` items to the available-space list. It should use memory allocation call only once.

5. A typical maze problem, where the goal is to find a path between two cells in a maze, is often solved with a stack. Answer the following questions:
   (3%) (a) What is the purpose of using a stack?

   (3%) (b) If using a stack, what information should be pushed onto the stack when moving to a new cell?

   (4%) (c) Can you use recursion to solve the same problem? Explain.

6. (10%) Briefly describe “binary search” and “searching on a binary search tree”, and make a comparison between them in terms of data structure constraints and time complexity.

7. (10%) Explain why a heap is always implemented in an array rather than linked list.

注意:背面有試題

8. (5%) (a) Propose two graph storage structures for the graph in Figure 1 and explain the pros and cons for those two structures.

   (5%) (b) Develop a minimal spanning tree of the graph.

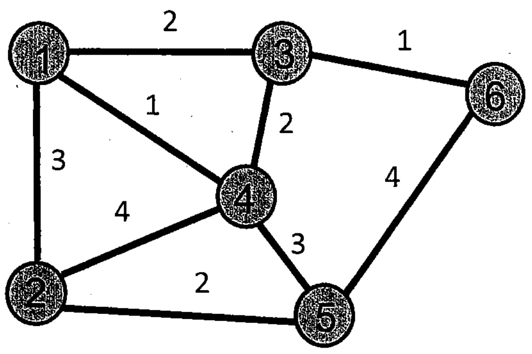

Figure 1. Graph with weights.

9. (10%) The binary search tree in Figure 2 was created by starting with a null tree and entering data from the keyboard. In what sequence were the data entered? If there is more than one possible sequence, identify the alternatives.

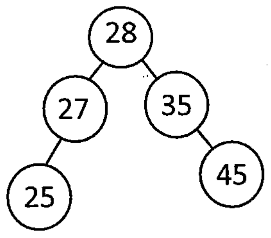

Figure 2. Binary search tree.

10. (10%) The shell sort algorithm is an improved version of the straight insertion sort. In the shell sort, a list of N elements is divided into K segments, where K is known as the increment. For example, in Figure 3, K is 3. The first, fourth, seventh, and tenth elements make up segment 1. For each pass, the data in each segment are sorted in **Insertion Sort**. Thus, after insertion sort, there are three different ordered lists. After each pass through the data, the increment is divided by two (K/2) until, in the final pass, it is 1. Please prove (explain) the time complexity of Shell sort is close to O(N log N).

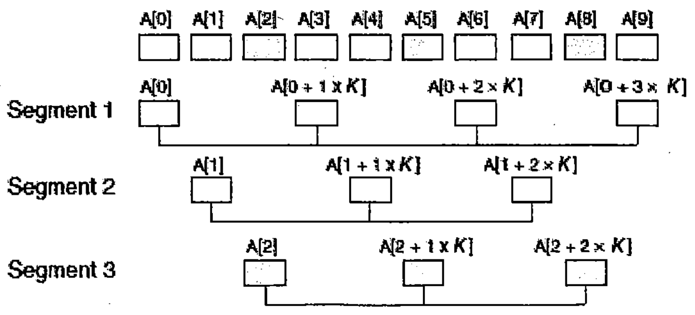

Figure 3. Segments in Shell sort.