國立交通大學108學年度碩士班考試入學招生試題

科目：計算機概論（5081）　　考試日期：108年2月14日 第2節

系所班別：資訊管理研究所碩士班　　組別：資管碩乙組　　

【不可使用計算機】＊作答前請先核對試題、答案卷（試卷）與准考證之所組別與考科是否相符!!

1. (15%) Please explain the following terms

(a) Proof of work
(b) Mashups
(c) BYOT
(d) NFC
(e) LSTM

2. (10%) List the types of information obtainable from data mining

3. (8%) Explain the logic concept of deep learning by examples

4. Artificial intelligence may be the most rapidly growing technology, and will change every aspect of our lives. Deep learning architectures such as deep neural networks and recurrent neural networks have been widely applied to various fields including computer vision, natural language processing, machine translation, bioinformatics, and board game programs, where they have produced results superior to human experts. In the research field of data science, please briefly explain the following methods or items:

(a) (4%) generative adversarial network
(b) (4%) reinforcement learning
(c) (4%) ensemble learning

5. In a disk scheduling system with 300 tracks indexed from 0 to 299, the disk head is currently located at track 38 and moving toward track 0. Now, a series of track access requests at track 39, 20, 241, 136, 299, 0 are arriving. Please briefly describe the following disk scheduling algorithms and calculate the total disk head traveling distance for above track access requests.

(a) (4%) SSTF
(b) (4%) SCAN
(c) (4%) LOOK

6. Please design two functions to solve greatest common divisor problem.

(a) (4%) int recursive_GCD(int m, int n)
this function uses the recursive algorithm, named as Euclidean algorithm (輾轉相除法). For example, recursive_GCD(64,18) is equal to recursive_GCD(18,10).

(b) (4%) int iterative_GCD (int m, int n)
this function uses iterative algorithm, which repeatedly executes a series of codes without any additional function call.

7. (5%) What are the four necessary conditions for characterizing deadlock? What's the difference between "deadlock prevention" and "deadlock avoidance" in operating system design?

8. (9%)

A priority_queue S is a data structure for maintaining a set of elements with the data type item, each with an associated key. A priority queue supports the following operations.

S.insert(x, k) inserts the element x with key k into the priority_queue S.

S.minimum( ) returns the element of S with the smallest key.

S.extract-min( ) removes and returns the element of S with the smallest key.

Show how to implement the push, pop and top operations of a stack with the operations supported by a priority queue.

```
class Stack{
 // variable count is used to keep track of the number of elements in the stack
 // You may need to use another variable and count to define the key for the priority queue.
 int count = 0;
 priority_queue S;
public:
   void push(item x);
   item pop( );
   item top( );
};
```

9.

A Linked Binary Search Tree List, BST-List, is a linked list which links an ordered sequence of binary search trees, $\mathbf{BST}_1$, $\mathbf{BST}_2$, …,$\mathbf{BST}_p$, …. A BST-List is defined as the following. (i) A BST-List contains two kinds of nodes, List-nodes and BST-nodes. List-nodes themselves form a singly linked list structure $\mathbf{List\text{-}node}_1$, $\mathbf{List\text{-}node}_2$, …, $\mathbf{List\text{-}node}_p$, …. Each group of BST-nodes forms a binary search tree BST_p. Each $List\text{-}node_p$ contains a pointer to the first node of a binary search tree BST_p . (ii) The key value of $List\text{-}node_p$ is less than the key value of $List\text{-}node_q$ in BST-List, for $p < q$. Furthermore, the key values in BST_p are less than the key values in BST_q, for $p < q$. (iii) Every element has a unique key value. Each BST-node stores pointers, an element's data and key value. List-nodes are only used to store key values and pointers.

(a) (5%) Define your data structures for List-nodes and BST-nodes. Draw an example diagram of your BST-list. What key values should be stored in List-nodes in order to facilitate an efficient search on your BST-list?

(b) (16%) Write an algorithm to search an element in a BST-list. Assume that the number of List-nodes is n and the maximum number of BST-nodes in a BST is m. Analyze the time complexity of your algorithm.