中原大學 102 學年度 **碩士班** 入學考試

102/3/2 13:30 ~ 15:00 資訊工程學系

誠實是我們珍視的美德，
我們喜愛「拒絕作弊，堅守正直」的你！

科目：資料結構與演算法 （共 6 頁第 1 頁）

□可使用計算機，惟僅限不具可程式及多重記憶者 ■不可使用計算機

注意事項：

1. **請勿在本試卷上直接作答!**
2. **請務必在答案卷上按照順序答題，以避免誤改！**
3. **每題都必須要有個題號！**
4. **即使你答不出來也必須要寫題號！答不出來就留空白。**
5. **問答題請盡量用中文句子來寫你的答案 (但專有名詞例如 queue、stack、tree、node、array index、graph 可以用英文)！**
   若你用英文句子回答，改卷者將不會去猜你到底在說什麼。

(1) (10 pts.)

(a) (2 pts.) A B-tree is a balanced search tree. What does “balanced” mean ?

(b) (2 pts.) Describe a case in which the use of B-trees (for storing keys) is desirable. Please also explain why the use of B-trees (for storing keys) is desirable for such cases.

(c) (6 pts.) When a new key is inserted onto a B-tree, the B-tree will “grow” as a result. Sometimes, this means that the B-tree will grow “taller.” Explain why a B-tree can *always* remain to be balanced after inserting a new key onto this B-tree.

(2) (10 pts.)

(a) (5 pts.) Describe the notion of abstract data types (ADT) (i.e., what is an ADT?). In your description, you must use an example to illustrate. Note: the example you give in your answer will ***not*** be considered a description of what an abstract data type is. The example you give in your answer will ***only*** be considered as a way of showing how the various requirements (of being an ADT) can be satisfied.

(b) (5 pts.) Suppose you are writing the code of an ADT. There will be data or procedures (in this code) that you want to "export" (i.e., allowing users of this ADT to access or call). There may also be data or procedures (in this code) that you want to "hide" (i.e., disallowing users of this ADT to access or call). How do you allow users of this ADT to access/call data/routines that you want to export? How do you prohibit users of this ADT from accessing/calling data/routines that you want to hide?

Note: In specifying the coding principles you use (to allow or disallow users of your ADT to access/call data/routines in your ADT code), you can assume the language you use (for implementing this ADT) is C, C++, or Java. However, you must first say what the language you use is before you specify your coding principles. (e.g., you first say "我用的是 Java.")

(3) (30 pts.)

We want to define an ADT. The ADT we want to define is "queue of integer." We want to use a ***circular array*** in our implementation. Please write the *complete* code of this ADT (just assume that you are writing the full content of a .c or .cpp or .java file). You can use any one of these three languages in writing your code: C, C++, or Java. However, you must first say what this language is before you write your code. (e.g., you first say "我用的是 Java.")

For ease of grading, we assume that the size of the array to use (in implementing a queue of integers) is 50. If an error case occurs, just print something feasible to the standard output (this is not the right way to do things in the actual practice; but it is acceptable for exams).

If you are using C in your implementation, this is how your ADT may be used (by users of this ADT). Your code needs only to support such usages.

```
# include “MyQueue.h”// Your ADT code is in MyQueue.c or
MyQueue.cpp
...
InitQueue() ;
if ( QueueEmpty() ) ... ;
if ( QueueFull() ) ... ;
Enqueue( 3 ) ;
int a ;
a = Dequeue() ;
```

If you are using C++ in your implementation, this is how your ADT may be used (by users of this ADT). Your code needs only to support such usages.

```
# include “MyQueue.h”// Your ADT code is in MyQueue.cpp
...
MyQueue * queue = new    MyQueue ;
if ( queue->Empty() ) ... ;
if ( queue->Full() ) ... ;
queue->Enqueue( 3 ) ;
int a ;
a = queue->Dequeue() ;
```

If you are using Java in your implementation, this is how your ADT may be used (by users of this ADT). Your code needs only to support such usages.

```
import MyADT.MyQueue ; // Your ADT code is in MyQueue.java
...
MyQueue queue = new    MyQueue() ;
if ( queue.Empty() ) ... ;
if ( queue.Full() ) ... ;
queue.Enqueue( 3 ) ;
int a ;
a = queue.Dequeue() ;
```

Note: Please be very careful in writing QueueEmpty() and QueueFull() (or Empty() and Full() in the case of C++ and Java). **If your code for QueueEmpty() and QueueFull() (or Empty() and Full() in the case of C++ and Java) does not make sense, you lose at least 15 points.**

(4) (6 pts.)

Complete the following recursive algorithm for computing the ***binomial coefficient***, i.e., $\binom{n}{k}$.

```
int binomial ( int n, int k )
    if ( n = = k || k = = 0 )   return _____;
    else                        return _____;
```

(5) (6 pts.)

Solve the following recurrences:

(a) $T(n) = 2\,T(n/2) + n^2$

(b) $T(n) = 4\,T(n/2) + n^2 \lg n$

(6) (28 pts.)

The **0-1** ***Knapsack*** problem is a well-known problem in the study of computer algorithms. The problem can be described as follows:

A thief robbing a store finds n items. The i-th item is worth v_i dollars and weight w_i pounds. If the thief wants to take as valuable a load as possible, but he can only carry W pounds in his knapsack. Each item must either be taken or left behind.

(a) (2 pts.) If the ***brute-force*** approach is used to solve the problem, what's the running time in asymptotic Θ-notation?

(b) (6 pts.) Let x_i denotes if the i-th item is taken or left behind (i.e., 1 or 0), the problem can be formulated mathematically as:

Maximize: _____

Subject to: _____

$x_i = 0$ or 1 for $i = 1 \ldots n$.

This is also known as the ***linear programming***.

(c) **(6 pts.)** Suppose the ***Dynamic Programming*** (DP) is used to solve the problem and let $c[i, w]$ be the total value of solution for the sub-problem (i.e., for the items $1 \ldots i$ and knapsack w), the algorithm can be designed as:

$$c[i,w] = \begin{cases} \_\_\_\_\_ & if\ i = 0\ or\ w = 0 \\ \_\_\_\_\_ & if\ w_i \geq w \\ \_\_\_\_\_ & if\ w_i < w \end{cases}$$

(2 pts.) What's the running time of the DP algorithm in Θ-notation?

(d) **(6 pts.)** If instead, the ***greedy algorithm*** (GA) is used to solve the problem, briefly explain the approach you use and design the algorithm using Pseudocodes or C/C++.

(2 pts.) What's the running time of the GA algorithm in asymptotic Θ-notation?

(e) **(4 pts)** Following from (d) and given a list of items in the following:

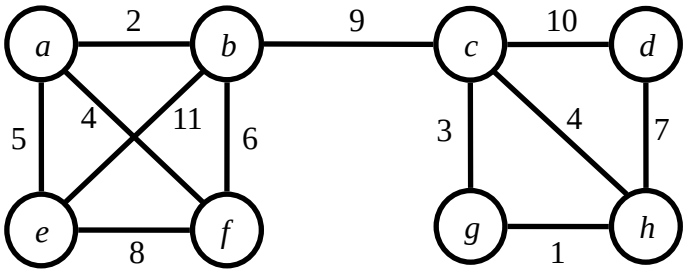
i	**1**	**2**	**3**	**4**	**5**	**6**
Value	25	20	45	5	15	40
Weight	10	10	15	5	10	10

Note: The unit of value is in US$, and the unit of weight is in pounds.

If the thief can only carry 40 pounds (or $W = 40$) in his knapsack, which items should he take? Is your solution optimal?

(7) (10 pts.)

Given the following graph, please answer the following:



(a) (3 pts.) If the ***Breadth-First Search*** (BFS) is used to search the graph, determine the BFS sequence.

(b) (3 pts.) If the ***Depth-First Search*** (DFS) is used to search the graph, determine the DFS sequence.

(c) (4 pts.) If the ***Prim's Algorithm*** is used to find the minimum spanning tree of the graph, illustrate the execution of the algorithm step by step. What's the sequence of the vertex being added to the tree if the root vertex is selected as the vertex *a*.

Note: For the BFS and DFS, if more than one vertex can be visited next at any moment, always visit the vertex in alphabetic order.