※ 注意：請用 2B 鉛筆作答於答案卡，並先詳閱答案卡上之「畫記說明」。

(一)是非題 (共 10小題，若你覺得該小題命題正確，請填 (A)，錯誤，請填 (B)。每題答對得 4 分，答錯倒扣 4 分，未填答者不計分也不扣分，倒扣至是非題總分為0分為止)

1. If a program has the computational complexity `Ω(X(n))`, where `X` is an exponential function of the program input size n, then the program must also have the computational complexity `O(X(n))`.

2. If we implement a *min-heap* using a *dynamic array*, then its `findMin()`, `insert()`, and `deleteMin()` operations have the time complexities `O(1)`, `O(log n)`, and `O(log n)`, respectively.

3. Since a *Deap* contains a subtree of *min-heap* and a subtree of *max-heap*, its `find()`, `deleteMin()`, and `deleteMax()` operations can all have `O(log n)` time complexities.

4. If we perform the post-order traversal from the *root* of a *tree* and mark the visited nodes with ascending numbers, then the root will have the largest number.

5. Assume each data access of a hard drive can fetch a "block" of K data, where K is usually a large number, say, 1024. If we store N (e.g. N = 10 millions) data in this hard drive using a *B tree* of order K, then the time to "find" a specific data from this hard drive is approximately $\log_K N$ * t, where t is the disk access time.

6. Since the leaf nodes of a *red-black* (*RB*) *tree* are all black nodes, when we insert a new data to a *RB tree*, we first insert this new data to the *RB tree* like a regular *binary search tree* operation (i.e. push it to the leaf), paint the corresponding new leaf node red, and then perform rotation(s) to balance the *RB tree*, if necessary.

7. Given a *strongly connected graph* (i.e. a tree with only one *strongly connected component*), we can find a trace that traverses from a starting vertex, goes through each edge exactly once, and terminates at the starting vertex.

8. A *stable set*, or *independent set*, of a *graph*, is a subset of vertices with the property that no two vertices in the stable set are adjacent. The *stability number* α(G) of a *graph* G is the cardinality of the largest *stable set*. For the classic *coloring problem* where the colors of any pair of the adjacent vertices must be distinct, we can solve it by partitioning the *graph* into *stable sets*. Therefore, the *Chromatic number* χ(G) of a graph G, that is, the minimum number of colors needed to color the *graph*, must be smaller than or equal to the *stability number* α(G) of the *graph*.

9. A *cube* can be represented as a *planar graph* and also a *bipartite graph*.

10. A perfect *hash function* is a function that can transform the key of a data into an integer between 0 and B - 1 with equal probabilities, where B is the number of buckets in the *hash*. Therefore, for a *hash* with a perfect *hash function*, its

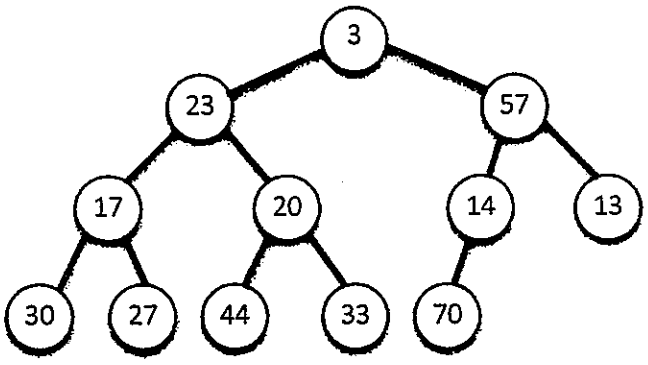
見背面

`find()`, `insertData()`, and `deleteMin()` have the time complexities `O(1)`, `O(1)`, and `O(log B)`, respectively, assuming `B` is large enough so that collisions seldom occur.

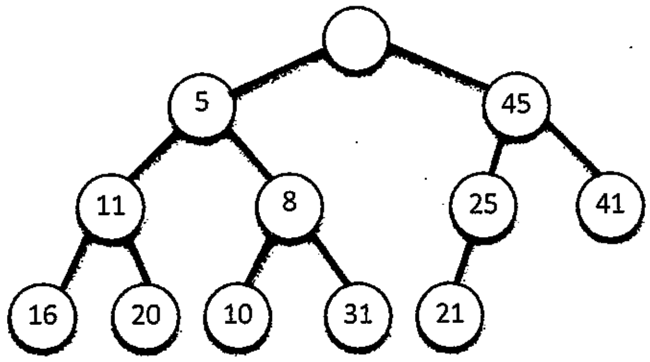
(二)複選題　(共四小題，每題 15 分, 題內每個選項單獨計分，答對得 3 分，答錯倒扣 3 分，倒扣至複選題總分為 0 分為止。如某題未作答則該題得 0 分，不倒扣。)
(例：正確答案為 ABC，若答 BCD，則答對三個選項 (BCE)，答錯兩個選項 (AD)，故共計得 3*3 - 3*2 = 3 分)

11. There are various types of *heap* data structure. Which of the following(s) is (are) a legal *heap* with respect to the typename shown beside the figure?

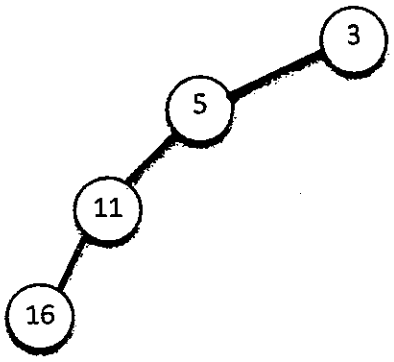
(A) *Min-Max Heap*
(B) Deap



(C) Leftist Heap

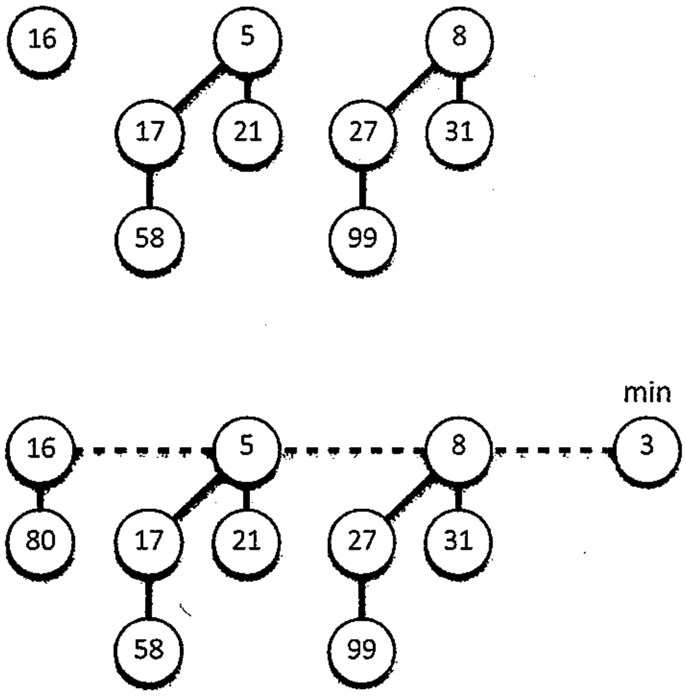


(D) *Binomial Heap*



接次頁

(E) *Fibonacci Heap*



12. Which of the following statement(s) about the *tree* data structure is (are) correct?

(A) (Number of edges) = (Number of nodes) - 1

(B) (Number of leaf nodes) = (Number of internal nodes) + 1
// Note: Leaf nodes are those with no child node. Nodes that are not leaf nodes are internal nodes.

(C) (Number of paths from the root to the leaf nodes) is O(Number of leaf nodes).

(D) (Height of a *tree*) is Ω(log(Number of nodes)).

(E) (Number of subtrees) = O((Number of nodes)^2)
// A *subtree* T_s of a tree T is a *tree* formed by a subset of nodes in T and with the edges that connect these nodes.

13. Which of the following statement(s) about the *graph* data structure is (are) correct?

(A) Two *graphs* are *isomorphic* if and only if both *graphs* contain the same number of vertices and the same number of edges.

(B) All non-empty *graphs* must contain at least a *clique.*

(C) Depth-first traversal of an arbitrary graph has the time complexity $O(n)$, where n is the number of vertices in the *graph.*

(D) The longest simple path of an arbitrary *graph* has the length $\Theta(n^2)$, where n is the number of vertices in the *graph.*

(E) A complete graph has the number of edges $\Theta(2^n)$, where n is the number of vertices in the *graph.*

14. If we design a *hash function* `h(s)` as follows:

$$h(s) = (\sum_{i} s[i]) \% B$$

where `s` is a string, `s[i]` returns the ASCII code of the i-th character in the *string*, % is the modulo operator, and `B` is the number of buckets in the *hash*. In short, given a string as the key of the data, this *hash function* sums the ASCII codes of all the characters and then return an integer between 0 and B - 1 by the modulo operation.

The ASCII codes for the English letters are as shown below. Let the data to be inserted to the hash be { "USA", "MIT", "Cat", "Dog", "May", "Sam", "Bob", "Low", "Phd", "See" }, and the number of buckers be 5. Which of the follow bucket(s) has (have) collision(s)?

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109

n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

(A) 0
(B) 1
(C) 2
(D) 3
(E) 4

試題隨卷繳回