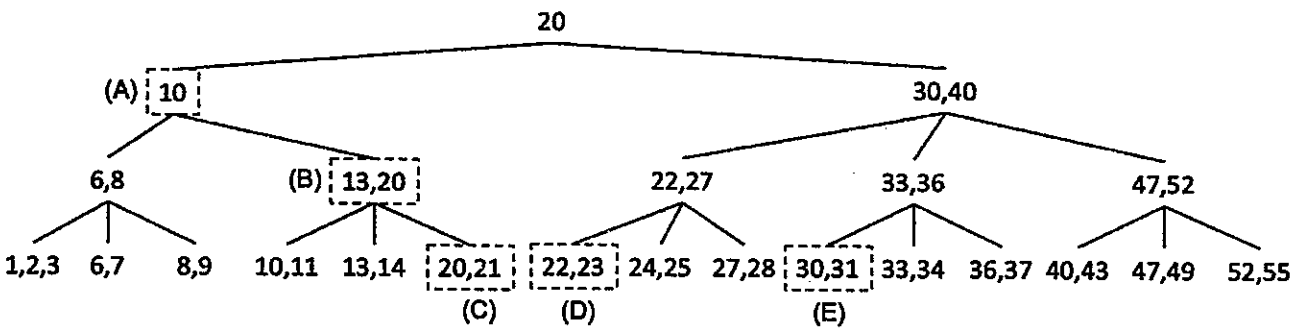
本試卷有 14 題選擇題(題號 1-14), 第 4,5,7 題為單選題, 其他選擇題為複選題。單選題該題答對得 5 分，答錯倒扣 1 分，未答者得 0 分。複選題每個選項單獨計分，選項答對者得 1 分，選項答錯者倒扣 0.5 分，另若該題整題未作答者，整題得 0 分。選擇題請於答案卡上作答。另有 2 題非選擇題(題號 15-16)，請於試卷本上作答。

Multiple Choice Problems (70 points, 5 points for each problem)

1. Assuming we have n data points. Among the variant characteristics of the data points, please select the correct descriptions.
   (A) If the worst-case running time is the most important, *merge sort* can be a good choice with $O(n \log n)$ time.
   (B) If the input happens to be sorted already, *bubble sort* can be a best choice with $O(n)$ time.
   (C) If the input array is in random order and the average sorting time is most important, *quick sort* can be a good choice with $O(n \log n)$ time.
   (D) If the exchanges of the items in the array are very expensive, *selection sort* will incur the least "swaps" or "moves".
   (E) If the input array consists of integers in the range $1...n^k$, *radix sort* with radix n with $O(k \log n)$ time.

2. B+ tree is an extension of B tree. The major differences from B tree are (1) all leaf nodes are linked together in a doubly-linked list, and (2) data points are stored on the leaf nodes only; internal nodes only hold keys and act as routers to the correct leaf node; the left child is smaller than the key and the right child is larger or equal than that. Please find any/all violations of a B+ tree structure in the following diagram. Assume the tree node can at most contain 4 data points (keys).



見背面

3. Please find the following table for the characters and their corresponding occurring probabilities. Please design a Huffman encoding tree and select the correct descriptions.

Symbol (X)	A	B	C	D	E	F	G
Prob (X)	0.15	0.06	0.24	0.21	0.09	0.21	0.03

(A) The codeword length for symbol C is 3
(B) The codeword length for symbol G is 5
(C) The codeword length for symbol E is 5
(D) The codeword length for symbol A is 4
(E) none of above is correct

4. （單選）We now use several algorithms to traverse a binary tree. Assuming there are a total number of N nodes, how many of the following statements about the **worst-case space complexity** are **TRUE**?

- Using DFS to traverse a balanced binary tree takes $O(N)$.
- Using DFS to traverse a binary tree takes $O(N)$.
- Using BFS to traverse a balanced binary tree takes $O(N)$.
- Using BFS to traverse a binary tree takes $O(N)$.

(A) 0　(B) 1　(C) 2　(D) 3　(E) 4

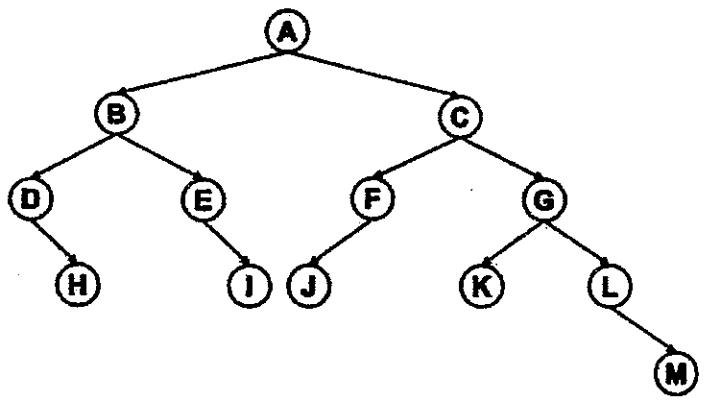
5. （單選）In a traditional merge sort, two sorted sub-arrays are combined to form a single, fully sorted array. This is referred to as a *2-way merge*. The problem at hand is to extend this concept by merging N sorted arrays of integers, where N and M are given integers representing the number of arrays and the number of integers in each array, respectively. Please choose the correct **worst-case time complexity** of this N-way merge.

(A) $O(N \log M)$　(B) $O(NM \log M)$　(C) $O(N^2 \log M)$
(D) $O(NM \log N)$　(E)none of the above is correct.

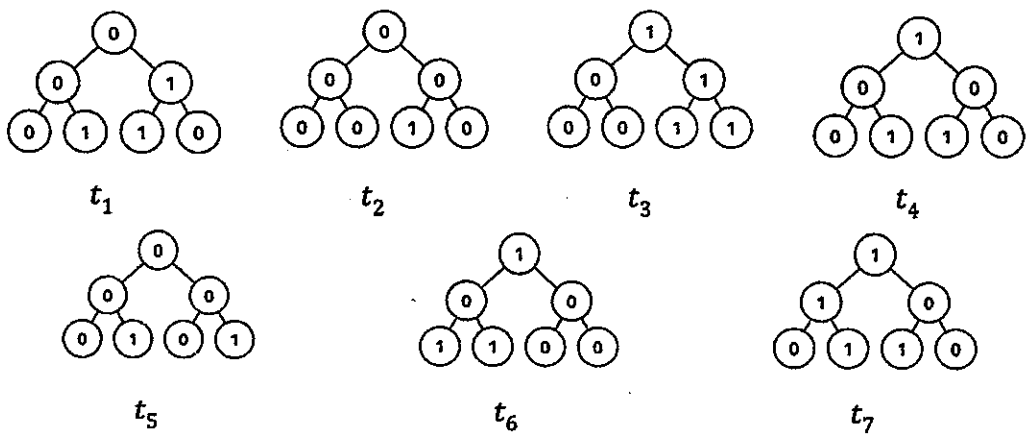
6. The following is a binary tree and the alphabet on the node is simply the "name" (instead of value) of the node. Please select the correct statements from the following:

(A) The successor of node B is E.
(B) The successor of node A is C.
(C) The tree is not a AVL tree.
(D) If we remove node J, the result is an AVL tree.
(E) If we remove node H, the result is an AVL tree.

接次頁



7. （單選）Given a list of binary trees $T = \{t_1, t_2, \ldots, t_7\}$ where each node is 0 or 1 shown as below, we would like to insert these trees into a **linear-probing hash table** of length $N = 11$. The hash function $f(t) = g\big(h(t)\big) \bmod N$, where $h(t)$ is the binary sequence obtained from *in-order* traversal of tree t and $g(\cdot)$ converts a binary sequence to a decimal number. For instance, $g(\text{"1111111"}) = 127$. Here's the question: how many collisions occur during the insertion process?



(A) 0　(B) 1　(C) 2　(D) 3　(E) 4

8. **Postfix advantages**: What is/are the advantage(s) of using postfix notation for math expressions?
   (A) No need to use parentheses
   (B) No need to consider precedence of operators
   (C) Easier for human to read
   (D) Easier for computers to evaluate
   (E) More concise when compared with prefix notation

見背面

9. **DFS advantages:** To find a feasible solution to the eight-queen problem, what is/are the reason(s) that we prefer to use DFS (depth-first search) instead of BFS (breadth-first search)?
   (A) DFS is more memory efficient.
   (B) DFS can be implemented using a stack.
   (C) DFS usually reaches the terminal states of "good" or "bad" faster.
   (D) DFS is conceptually simpler than BFS.
   (E) BFS cannot be implemented using a stack.

10. **Heap applications:** Which one(s) of the following applications may involve the use of heaps to increase time efficiency?
   (A) Find the k largest numbers from a given stream of numbers.
   (B) Given a stream of numbers coming one after another, calculate the median of the currently received set of numbers.
   (C) Compute the page rank of a given set of web pages
   (D) Detect appropriate "buy" and "sell" requests to create transactions in stock market
   (E) Identify the next timing for collisions in event-driven simulation for molecular dynamics

11. **Properties of dynamic programming (DP):** Which of the following statements is/are correct about DP?
   (A) Any DP problem can be visualized as the optimal path finding problem.
   (B) Once a DP problem is solved, all the related sub-problems are also solved.
   (C) Once a DP problem is solved, it is straightforward to obtain the second-best solution.
   (D) The optimal solution of a DP problem can be obtained using optimal solutions of its sub-problems.
   (E) A DP problem has overlapping sub-problems which are reused several times when solving the original problem.

12. **Properties of shortest path problem:** Which of the following statements is/are correct?
   (A) Dijkstra's algorithm can be applied to any directed graph with no negative cycle.
   (B) The heap data structure is likely to be used in Dijkstra's algorithm.
   (C) Floyd-Warshall algorithm can be applied to any directed graph with negative weights.
   (D) Floyd-Warshall algorithm is based on the concept of dynamic programming.
   (E) Every shortest path in a weighted directed graph G will not change if an extra weight is added on every edge of G

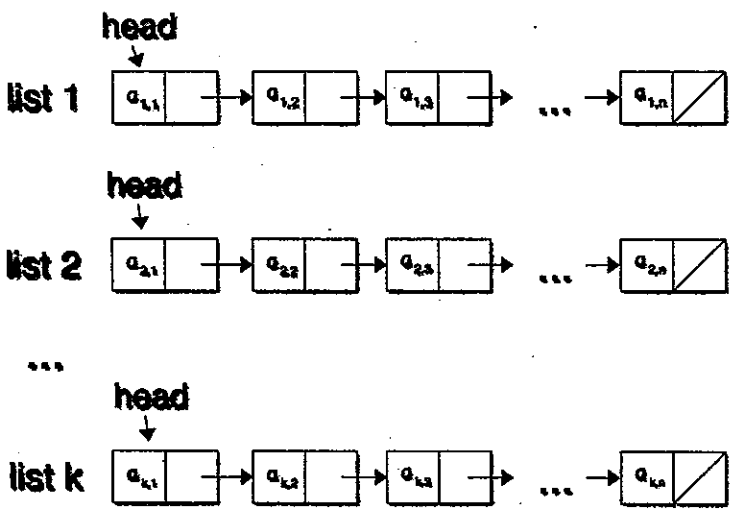
接次頁

13. **Properties of min. spanning tree (MST)**: Let T be a MST of a weighted graph G with at least 3 vertices. Which of the following statements is/are correct?
    (A) Given an edge e in G but not in T, we can form a cycle by putting e to T. Then e has the largest weight among edges in C.
    (B) If we partition G into two subsets, and let e be the smallest-weight edge across the partition. Then e belongs to some MST in G.
    (C) The edge with the smallest weight in G must belong to some MST of G.
    (D) The edge with the second smallest weight in G must belong to some MST of G.
    (E) The edge with the third smallest weight in G must belong to some MST of G.

14. **Prefix, infix, and postfix**: Which of the following statements is/are correct?
    (A) We can use a stack to convert an infix to postfix expression.
    (B) We can use a stack to convert a postfix to an infix expression.
    (C) We can derive a binary tree from its infix and prefix notations.
    (D) We can derive a binary tree from its infix and postfix notations.
    (E) We can derive a binary tree from its prefix and postfix notations.

Short Answer Problems (30 points)

15. **(21 points)** Given k singly linked lists, each of which has n nodes. The numbers in the nodes of the i-th list are given by $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$,as shown in the figure below. Each of the k lists has the numbers sorted in non-decreasing order, *i.e.*, $a_{i,1} \leq a_{i,2} \leq \cdots \leq a_{i,n}$, where i is the index of the list. Professor Q asks the students to develop an algorithm to merge these k lists into one singly linked list sorted in non-decreasing order. Three students have come up with different algorithms, which are given below in pseudo code.



見背面

Student A:

L1. Create an empty singly linked list *S* to collect the result.
L2. DO
L3. Iterate over the numbers in the head nodes of the *k* lists, and find the node with the smallest number, denoted as node *M*.
L4. Remove node *M* from the original list, and insert it into the result list *S* from the tail.
L5. UNTIL all original *k* lists are empty
L6. Return the result list *S*, which contains all nodes from the original *k* lists and sorted in non-decreasing order.

Student B:

L1. Create an empty singly linked list *S* to collect the result.
L2. Create an empty min heap *H*.
L3. Remove the *k* head nodes from the *k* lists and insert them into *H*. Each node in *H* also contains the index number *i* of the list where it is from.
L4. DO
L5. Remove the min node *M* from the heap. Insert this node into *S* from the tail. Take note of the index number stored in *M*, denoted as *m*.
L6. If list *m* is not empty, remove the head node from list *m* and insert it into *H*.
L7. WHILE *H* is not empty
L8. Return the result list *S*, which contains all nodes from the original *k* lists and sorted in non-decreasing order.

Student C:

Call the recursive divide-and-conquer function MergeTwoGroupLists (*k sorted linked lists)*. The return value of the function is a list containing all nodes from the original *k* lists and sorted in non-decreasing order.

L1. Function MergeTwoGroupLists (*m* sorted linked lists)
L2. If *m==1* then return the only list from the input.
L3. Divide the input *m* lists into two groups of lists, with $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$ lists, respectively.
L4. Recursively call MergeTwoGroupLists for each of these two groups of lists, merging the first group of the lists into one sorted list, S_1 , and the second group of the lists into one sorted list, S_2.
L5. Merge the two sorted lists S_1 and S_2 into one sorted list *S*.
L6. Return *S*.

For each of these three algorithms, analyze and give their asymptotic worst-case running times with **the big-O notation** and **in terms of *k* and *n***.

接次頁

Note:

- Lower-order terms and constant factors (excluding k and n) should be removed in the big-O notation in the answer.
- The bound in the answer needs to be tight.
- Only the final result will be graded and only fully correct answers will be given points. Please clearly mark your final result in the answer sheet.

16. **(9 points)** The pseudo code function to compute the prefix function in the Knuth-Morris-Pratt (KMP) string matching algorithm is given below. Given a pattern string $P[1..m]$, the prefix function for this pattern P is the function $\pi: \{1,2,\ldots,m\} \rightarrow \{0,1,\ldots,m-1\}$ such that $\pi[q]$ is the length of the longest prefix of P that is a proper suffix of P_q, where P_q denotes the q-character prefix of the string P. For pattern string P=ABACABACABACABAD, how many times is **line 7** executed? (Note: only the final answer will be graded and only fully correct answer will be given points)

```
COMPUTE-PREFIX-FUNCTION(P)
 1  m = P.length
 2  let π[1..m] be a new array
 3  π[1] = 0
 4  k = 0
 5  for q = 2 to m
 6       while k > 0 and P[k + 1] ≠ P[q]
 7            k = π[k]
 8       if P[k + 1] == P[q]
 9            k = k + 1
10       π[q] = k
11  return π
```

試題隨卷繳回