※ 考生請注意：本試題 □可 ☑不可 使用計算機

1. (24%) Please answer the following MULTIPLE-CHOICE questions about data sorting, insertion, deletion, searching and rotation. NOTES: Some have more than one correct answer to select. Negative scores will be applied if all guessing wrong.

(1) About the insertion sort or INSERTION-SORT(A) in pseudocode:
(a). it is an efficient algorithm for sorting a small number of elements.
(b). a sequence of n numbers $<a_1, a_2, \ldots, a_n>$ are sorted in place within an array.
(c). the time taken by this sort procedure depends on the output.
(d). it uses an incremental approach to design algorithms.
(e). to analyze only its worst-case running time is more appropriate than the other.

(2) About the merge sort or MERGE-SORT(A, p, q, r) in pseudocode:
(a). it runs faster than the insertion sort for all cases.
(b). it takes the running time $\Theta(1)$.
(c). its worst-case running time is $\Theta(n\lg n)$, when $|A|=n$.
(d). it closely follows a three-step divide-and-conquer approach to design algorithms.
(e). the key operation of its algorithm is the merging of two sorted sequences in the first step of the divide-and-conquer approach.

(3) About the heap, heapsort or HEAPSORT(A, i) in pseudocode:
(a). its running time is similar to insertion sort.
(b). its sorting approach is similar to merge sort which uses divide-and-conqure.
(c). the most popular applications of a heap is its use as an efficient priority queue.
(d). the basic procedures used in a priority-queue data structure, such as HEAP-MAXIMUM procedures, run in $O(n\lg n)$ time, when $|A|=n$.
(e). using either bottom-up or top-down version of heapify function, the heapsort algorithm is overall $O(n)$.

※ 考生請注意：本試題 □可 ☑不可 使用計算機

(4) About the quicksort

(a). its key procedure is PARTITION.

(b). its expected running time is O(lg$n$).

(c). any proportional split produced by the partition allows quicksort to yields an O(lg$n$) running time.

(d). when using RANDOMIZED-PARTITION, the expected running time of quicksort is also O($n$lg$n$)

(e). when partitioning to produce good and bad splits, the running time of quicksort is always like the running time of the bad splits with a slightly larger constant hidden by the O-notation.

(5) Please select the correct statement(s) listed below:

(a). there is no worst-case time to a perfect hashing table.

(b). it takes O(lg$n$) time to insert or to delete a node for an n-node red-black tree.

(c). when rotating an n-node red-black tree, either left or right, it takes O(1) time.

(d). without comparing keys in a binary search tree, we cannot determine both the successor and predecessor of a node.

(e). it can be made to run in O(h) time, either inserting or deleting an element in a binary search tree of height h.

(6) Which statement(s) is/are **true** listed below:

(a). for each node x in an n-node AVL binary search tree, the heights of the left and right subtrees of x differ by at most 1.

(b). a recursive procedure AVL-INSERT on an n-node AVL tree takes O($n$lg$n$) time and performs O(1) rotations.

(c). on an n-node splay trees, the amortized cost of each operation is O($n$lg$n$).

(d). each operation runs in expected O($n$lg$n$) time on a skip list of n items.

(e). the expected height of a randomly built binary search tree on n keys is O(lg$n$).

2. (20%) Please create a table, a graph or other formats you think it is proper to differentiate the following eight different sorting techniques? And please explain clearly why you differentiate them in this way? (1)bubblesort, (2)counting sort, (3)heapsort, (4)insertion sort, (5)merge sort, (6)quicksort, (7)selection sort, (8)Shell's sort

3. (6%) Please exemplify a binary search tree satisfying five red-black properties (3%) and then write a pseudocode for its either left or right rotation (3%).

※ 考生請注意：本試題 □可 ☑不可 使用計算機

4. (28%) True or False, and EXPLAIN
Circle T or F for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why or give a counter example. Answers without reasons will get at most 1 point.

(a) (**T** , **F**) Given two $n \times n$ integer matrics A and B. It takes $\Omega(n^2)$ time to calculate their multiplication $A \cdot B$, and $O(n)$ time to calculate their addition $A + B$.

(b) (**T** , **F**) In a min-heap of n objects, to identify the object that has the 3rd smallest key takes $O(\log n)$ time.

(c) (**T** , **F**) A min-heap of $n \geq 3$ objects can never be a binary search tree, if all the keys are distinct to each other.

(d) (**T** , **F**) Given an undirected connected graph $G = (V, E)$, to check whether node A connects to both node B and node C can be done in $\Omega(|E|^2)$ time, since we should scan each arc at least once for $A \rightarrow B$ and for $A \rightarrow C$, too.

(e) (**T** , **F**) A red-black tree of 7 nodes may contain zero red nodes.

(f) (**T** , **F**) Suppse n keys are already stored in a binary search tree T. Then using the in-order tree walk on T to sort these n keys is more efficient than any comparison sort algorithm.

(g) (**T** , **F**) To conduct a bread-first-search on a red-black tree of n nodes takes $O(\log n)$ time.

5. [4%] Let A, B, C, D, E, F, G, H be eight data items with the following assigned weights:

data	A	B	C	D	E	F	G	H
weight	22	5	11	19	2	11	25	5

Construct an optimal binary tree using the algorithm by Huffman.

※ 考生請注意：本試題 □可 ☑不可 使用計算機

6. (18%) Let $G = (V, E)$ be an undirected public bus transit network in a metropolitan area, where each vertex $i \in V$ represents a bus stop and each undirected edge $(i, j) \in E$ denotes a route segment between bus stop i and j. Suppose there are totally R bus routes and each route is a simple path in G. For each undirected edge $(i, j) \in E$, let X_{ij} be the set of bus routes that passes both bus stops i and j. In other words, each bus in X_{ij} passes stop i and then stop j. For each edge $(i, j) \in E$, let c_{ij} denote the traveling distance between bus stop i and j, respectively. Suppose Tom plans to travel from bus stop s to bus stop t in G.

(a) [4%] Explain how to check whether Tom's travel plan is feasible or not (i.e. whether Tom can really travels from s to t via buses in G) in <u>linear-time</u> (in terms of $|E|$ or $|V|$). Explain your answer.

(b) [4%] Suppose Tom makes the trip $v_1 \to v_2 \to v_3 \to \ldots \to v_q$, which passes nodes v_i for $i = 1, \ldots, q$ in order.
A bus transfer happens when one takes off at bus stop v_i from bus stop v_{i-1} via a bus route $r_{i-1,i} \in X_{i-1,i}$, and then takes a different bus route $r_{i,i+1} \in X_{i,i+1}$ from v_i heading for v_{i+1}. Give an upper bound on the number of different paths that Tom may take for this trip $v_1 \to v_2 \to v_3 \to \ldots \to v_q$.
(hint: any route segment can form a complete bipartite graph; your answer should be some big-O notation in terms of $|X_{v_i v_j}|$, the size of $X_{v_i v_j}$, for some route segment $v_i \to v_j$).

(c) [10%] Following from (c), give a method to calculate the least number of bus transfers required in the travel $v_1 \to v_2 \to v_3 \to \ldots \to v_q$. Explain why your method works and give its complexity. (hint: following the hint of (b), assign suitable edge weights)