類組：電機類　科目：資料結構(3002)　　共 2 頁 第 1 頁

問答題

1. (10%)Please write a function to delete a selected node in a given singly linked list. For example, given a linked list: 1->3->4->2 and the selected node of 4, your function should return the list as 1->3->2.

2. (10%)Please write a function to swap the position m and n in a singly linked list, where $1 \leqq m \leqq n \leqq$ the length of the linked list. For example, given m=2, n=4, and a linked list: 1->5->4->3. Your function should return the list as 1->3->4->5. Please note that the list should only be traversed once (i.e., in one-pass).

3. Please convert an infix expression (a/b)+c-(f/d)*g to the postfix (5%) and prefix (5%) expressions by using a stack. Your answer should include the detailed input, stack, and output in each step.

4. (10%)For each of the four types of lists in the following table, what is the asymptotic worst-case running time for each dynamic-set operation listed?

	Unsorted, singly linked	Sorted, singly linked	Unsorted, doubly linked	Sorted, doubly linked
SEARCH(L,k)				
INSERT (L,x)				
Delete(L,x)				
MINIMUM(L)				
MAXIMUM(L)				

5. (10%) Is Heap sort stable? Justify your answer by giving examples.

6. (10%) In double hashing, we use a secondary hash function to produce different collision paths for different keys. Please give a design of double hashing and show how your design can avoid both primary and secondary clustering.

7. (10%) In what condition (worst case) the time complexity of quick sort will be $O(n^2)$? Please provide the detailed proof of this result. Under the same condition, what modification of quick sort can turn the worst case into the best case? What is the best-case time complexity?

8. (10%) There are two algorithms for minimum spanning trees (MSTs) for undirected graphs: Kruskal and Prim. If there exists more than one MSTs for a given graph, will these two algorithms find the same MST? If not, please give examples and explain why.

注意:背面有試題

9. (10%) Please design an efficient algorithm to find a path from u to v such that the biggest edge weight on that path can be minimized. In other words, if all the paths from u to v have their edge weights sorted (there should exist a biggest edge weight in all edges), this path found by the algorithm has the smallest "biggest-edge-weight" among all paths' biggest edge weights.

10. (10%) Binary search trees (BSTs)
    (a) What is the resulting BST from successively inserting the keys 7, 3, 6, 9, 8, 4, 2 into an initially empty tree?
    (b) What will be the resulting BST after we delete key 3 in the BST in (a)?