大同大學 101 學年度(寒)轉學入學考試試題

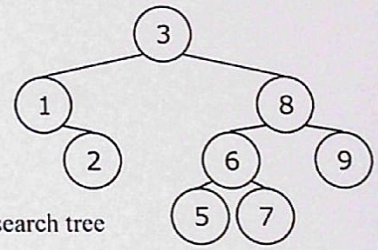
考試科目:資料結構　　所別:資訊工程學系　　

註:本次考試 不可以參考自己的書籍及筆記； 不可以使用字典； 不可以使用計算器。

I. Pick the correct answer for each question (20 points).

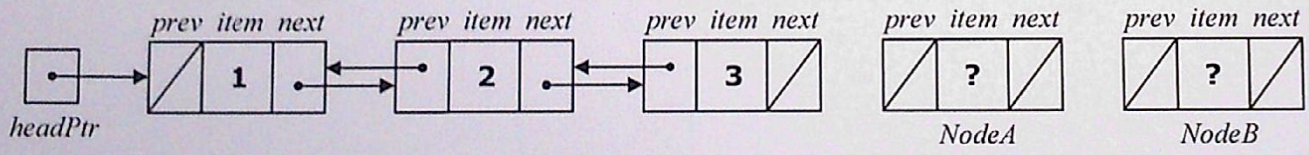
1. If the left subtree and right subtree of a nonempty binary tree T are T_L and T_R, then the number of nodes of T, count(T), is (1) count(T_L) + count(T_R) (2) max{ count(T_L), count(T_R) } (3) min{ count(T_L), count(T_R) } (4) 1 + count(T_L) + count(T_R) (5) 1 + min{ count(T_L), count(T_R) } (6) 1 + max{ count(T_L), count(T_R) }.
2. The maximum height of a binary search tree with 4 nodes is (1) 1 (2) 2 (3) 3 (4) 4 (5) 5.
3. The minimum height of a binary search tree with 4 nodes is (1) 1 (2) 2 (3) 3 (4) 4 (5) 5.
4. Which of the following trees may not be a height balanced tree if the tree has 5 nodes? (1) a 2-3 tree (2) an AVL tree (3) a max heap (4) a red-black tree (5) a binary search tree.
5. The best case running time of retrieving an item from a hash table with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2
6. The worst case running time of retrieving an item from a hash table with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2
7. The best case running time of retrieving an item from a search tree with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2
8. The worst case running time of retrieving an item from a search tree with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2
9. The best case running time of retrieving an item from an unsorted list with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2
10. The worst case running time of retrieving an item from an unsorted list with n items is **O**(?) (1) **1** (2) log n (3) n (4) n log n (5) n^2

II. Drawing (20 points)



11. (5 points) If a node with value 4 is inserted into the AVL tree on the right, the tree becomes unbalanced and needs rebalancing. Draw the resultant tree after rebalancing.
12. (5 points) If the node with value 2 is deleted from the AVL tree on the right, the tree becomes unbalanced and needs rebalancing. Draw the resultant tree after rebalancing.
13. (5 points) The values in the set { **5, 6, 4, 3, 7, 8, 2, 1, 9** } are inserted in sequence into a binary search tree which is initially empty. Draw the resultant tree after the values are all inserted.
14. (5 points) Draw the binary expression tree for the infix expression: **5 + 2 * 3 – 6 + 8 / 4 – 1 * 7**.

III. Programming (60 points)



15. (20 points) The above linked list has 3 nodes with item values 1, 2 and 3. The ***prev*** of a node points to the previous node in the list, the ***next*** of a node points to the next node in the list, and the ***item*** contains the numerical data of a node. The ***headPtr*** points to the first node of the list. ***NodeA*** and ***NodeB*** are two existing nodes. Write the C/C++ code that can function as requested below.
Note. DO NOT use any loop structures (***for*** or ***while***) in your code and DO NOT define any variables. That is, you CAN ONLY use the variables already in the above graph (e.g., ***NodeA. prev***, ***NodeB. next***, ***headPtr ✳ next ✳ prev***, etc.).
(1) (5 points) Insert ***NodeA*** between the nodes with item values 2 and 3.
(2) (5 points) Delete the node with item value 2.
(3) (5 points) Insert ***NodeB*** before the node with item value 1.
(4) (5 points) Delete the first node.
16. (10 points) Write an iterative C/C++ function, ***displayI***(char ***A***[], int ***n***), that will display the ***n*** characters in the array ***A*** in reverse order.
17. (10 points) Write a recursive C/C++ function, ***displayR***(char ***A***[], int ***n***), that will display the ***n*** characters in the array ***A*** in reverse order.
18. (10 points) Write an iterative C/C++ function, ***findMinI***(int ***A***[], int ***n***), that will return the minimum of the ***n*** numerical values in the array ***A***.
19. (10 points) Write a recursive C/C++ function, ***findMinR***(int ***A***[], int ***n***), that will return the minimum of the ***n*** numerical values in the array ***A***.

<<以下空白>>