

Homework 3

(Due on February 14, 2008)

1. Please select from one of the following answers for the questions listed below:
 - a. $O(1)$
 - b. $O(n)$
 - c. $O(\log n)$
 - d. $O(n \log n)$
 - e. $O(n^2)$
 - f. None of the above.

- (1) The time complexity of finding the maximum value in a single-linked list of integers sorted in ascending order. **(B)**
- (2) The time complexity of finding the maximum value in a single-linked list of integers sorted in descending order. **(A)**
- (3) The time complexity of finding the maximum value in a double-linked list of integers sorted in ascending order. **(A)**
- (4) The time complexity of finding the maximum value in an unsorted double-linked list of integers. **(B)**
- (5) The time complexity of removing the last element of a double-linked list without a tail pointer. **(B)**
- (6) What is the worst case time complexity of searching for an item in a sorted array using binary search? **(B)**
- (7) The time complexity of finding the minimum value in a complete binary minimum heap. **(A)**
- (8) The time complexity of inserting an element to a heap tree with n nodes. **(C)**
- (9) The time complexity of building a heap tree with n nodes by insertion method. **(D)**
- (10) The time complexity of giving a key to find a value stored in the hash. **(A)**

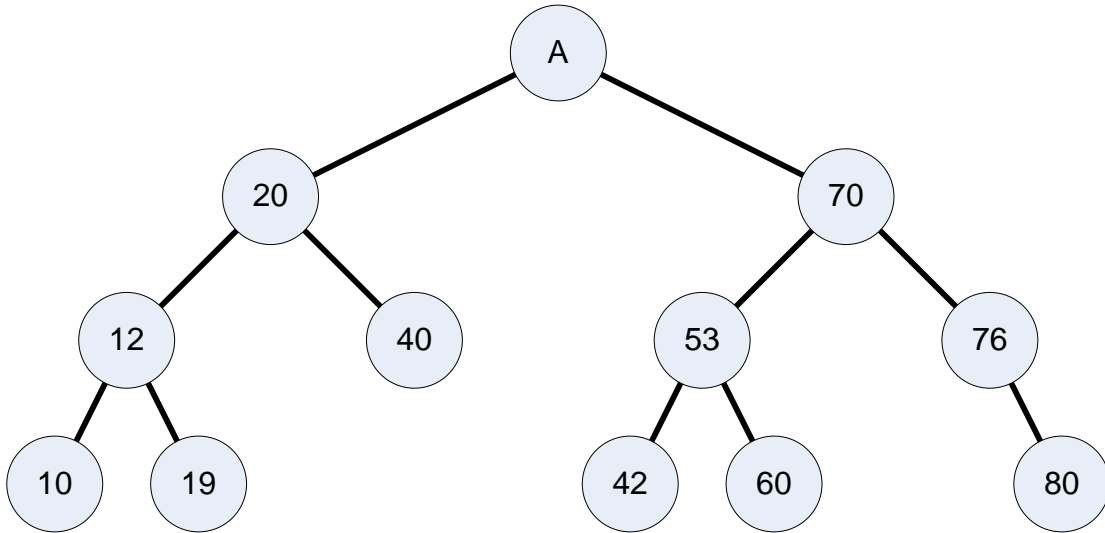
NOTE: For question (6), most students chose (C) which is the average case, the worst case is $O(n)$.

CS 303L Data Structure and Algorithm

February 7, 2008

2. Assume that a binary tree is built for storing a set of integer values as shown below:

Note: A set indicates a collection of distinct values.



- (1) What is (are) the possible value(s) for the root A? **(41)**
- (2) What is (are) the value(s) for the leaves? **(10, 19, 40, 42, 60, 80)**
- (3) If a new node with value 54 is inserted, where the node will be placed? **(Left child of 60)**
- (4) What will you obtain if performing the following operations? **(a: 19 / b: null)**
 - a. `root.getLeftNode().getLeftNode().getRightNode()`
 - b. `root.getRightNode().getRightNode().getLeftNode()`

CS 303L Data Structure and Algorithm

February 7, 2008

3. The following are some sorting algorithms:
 - a. Bubble Sort
 - b. Heap Sort
 - c. Insertion Sort
 - d. Merge Sort
 - e. Quick Sort
 - f. Bucket Sort

Please answer the following questions:

- (1) Which algorithm(s) mentioned above is (are) recursive algorithm? **(D,E)**
- (2) Which algorithm(s) mentioned above is (are) stable? **(A,C,D,F)**
- (3) Which algorithm(s) mentioned above is (are) not comparison sort? **(F)**
- (4) Which algorithm(s) mentioned above has (have) time complexity of $O(n \log n)$ in the worst case? **(B,D)**