

April 2013

Name

[40 points] Fill in the blank(s) for questions 1-23. Please write clearly; your answers to these 23 questions must be on the exam, not on a separate paper.

- 33
- The sum of the powers of two ($2^0 + 2^1 + 2^2 + \dots + 2^n$) is equal to $\frac{2^{n+1} - 1}{1}$.
 - The sum of increasing integers ($1 + 2 + 3 + \dots + n$) is equal to $\frac{(n+1)n}{2}$.
 - $\log_b b^x$ is equal to x .
 - $\log_b 1$ is equal to 0 .
 - $b^{\log_b b^x}$ is equal to (1) .
 - A logarithmic algorithm has a time complexity of $O(\log n)$.
 - A quadratic algorithm has a time complexity of $O(n^2)$.
 - A linear algorithm has a time complexity of $O(n)$.
 - A binary tree that has 10,000 internal nodes will have a height between 10.00 and 10000 ($h \leq 13$).
(Hint: $2^{13} = 8,192$ and $2^{14} = 16,384$).
 - A red-black tree that has 10,000 internal nodes will have a height between $\log 1000$ and 10000 . $\rightarrow 2^{13} - 12^3$
 - An AVL tree that has 10,000 internal nodes will have a height between $\log 1000$ and 10000 . $\rightarrow 2^{14} - 12^4$
 - Post-order traversal of a binary tree means the parent, left child, and right child nodes are "visited" in the following order left child, right child, and parent.
 - In-order traversal of a binary tree means the parent, left child, and right child nodes are "visited" in the following order left child, parent, and right child.
 - The maximum depth of any external node of a tree T is called the height of the node of T .
 - In a circular, growable array implementation of the Queue ADT, the enqueue and dequeue operations run in $\Omega(1)$ and $\Omega(1)$ amortized time if the array size is increased by a constant C each time it has to be enlarged.
 - In a growable array-based implementation of a Stack ADT, the push operation runs in $O(1)$ amortized worst-case time when the array size increases by double each time resizing is necessary.
 - In a hash table implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in 1 , 1 , and 1 expected time respectively.
 - In an unsorted, growable array implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in 1 , n , and 1 time respectively.
 - In a sorted, growable array implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in n , 1 , and n time respectively.
 $\log n$