

CS435 Algorithms Mid-Term exam

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
1. The sum of the powers of two ($2^0 + 2^1 + 2^2 + \dots + 2^n$) is equal to $\frac{2^{n+1} - 1}{1}$.
 2. The sum of increasing integers ($1 + 2 + 3 + \dots + n$) is equal to $\frac{(n+1)n}{2}$.
 3. $\log_b b^x$ is equal to x .
 4. $\log_b 1$ is equal to 0 .
 5. $b^{\log_b b^x}$ is equal to (b^x) .
 6. A logarithmic algorithm has a time complexity of $O(\log n)$.
 7. A quadratic algorithm has a time complexity of $O(n^2)$.
 8. A linear algorithm has a time complexity of $O(n)$.
 9. A binary tree that has 10,000 internal nodes will have a height between 14.00 and 10000 ($h \leq 14$).
(Hint: $2^{13} = 8,192$ and $2^{14} = 16,384$).
 10. A red-black tree that has 10,000 internal nodes will have a height between $\log 1000$ and 10000 . $\rightarrow 2^{13} - 12^3$
 11. An AVL tree that has 10,000 internal nodes will have a height between $\log 1000$ and 10000 . $\rightarrow 2^{14} - 12^4$
 12. 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.
 13. 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.
 14. The maximum depth of any external node of a tree T is called the height of the root of T .
 15. 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.
 16. 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.
 17. In a hash table implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in 1 , 1 , and 1 expected time respectively.
 18. In an unsorted, growable array implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in 1 , n , and 1 time respectively.
 19. In a sorted, growable array implementation of the Dictionary ADT, the insertItem, findElement, and removeElement operations run in n , 1 , and n time respectively.
 $\log n$