PRACTICE FOR TEST 1

Advanced Algorithms I (7081) Fall 2015

The questions on Test 1 on Wednesday, Oct 21 will be similar to a (small) subset of questions from this Practice Test.

Multiple Choice: For each of the following questions **circle** the **best** answer.

1. *Quicksort* has best, average and worst case complexities, respectively:
   a) $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n \log n)$
   b) $\Theta(n \log n)$, $\Theta(n \log n)$, $\Theta(n \log n)$
   c) $\Theta(n \log n)$, $\Theta(n \log n)$, $\Theta(n^2)$
   d) $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^2)$
   e) $\Theta(n^2)$, $\Theta(n^2)$, $\Theta(n^2)$
2. *Mergesort* has best, average and worst case complexities, respectively:
   a) $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n \log n)$
   b) $\Theta(n \log n)$, $\Theta(n \log n)$, $\Theta(n \log n)$
   c) $\Theta(n \log n)$, $\Theta(n \log n)$, $\Theta(n^2)$
   d) $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^2)$
   e) $\Theta(n^2)$, $\Theta(n^2)$, $\Theta(n^2)$
3. *Heapsort* has worst case complexity:
   a) $\Theta(\log n)$
   b) $\Theta(n)$
   c) $\Theta(n \log n)$
   d) $\Theta(n^2)$
   e) None of the above
4. An sorting algorithm that is **not** comparison-based:
   a) Mergesort
   b) Quicksort
   c) Insertion Sort
   d) Treesort
   e) Radix Sort
5. An algorithm with worst-case complexity $W(n)$ is said to be polynomial-time if
   a) $W(n)$ is a polynomial
   b) $W(n) \in O(n^k)$ for some constant k
   c) $W(n) \in \Omega(n^k)$ for some constant k
   d) $W(n) \in \Theta(n^k)$ for some constant k
   e) $W(n)$ has sub-exponential order
6. The smallest worst-case complexity of a comparison-based sorting algorithm is:
   a) $\Theta(\log n)$
   b) $\Theta(n)$
   c) $\Theta(n \log n)$
   d) $\Theta(n^2)$
   e) $\Theta(n^2)$
7. The smallest worst case complexity for computing the maximum and minimum elements in an list of even size *n:* i
   a) $n - 1$
   b) $3n/2 - 2$
   c) $2n - 2$
   d) $5n/2 - 1$
   e) n^2

8. Linear Search and Interpolation Search have worst-case complexities, respectively:
   a) $\Theta(\log n)$, $\Theta(1)$
   b) $\Theta(\log n)$, $\Theta(\log n)$
   c) $\Theta(\log n)$, $\Theta(n)$
   d) $\Theta(n)$, $\Theta(\log n)$
   e) $\Theta(n)$, $\Theta(n)$
9. Binary Search assumes as a precondition that:
   a) The search element is on the list
   b) The list consists of integers or floating point numbers
   c) The list satisfies a uniform distribution
   d) The list is sorted once the search element is added
   e) The list is sorted
10. $1 + 2 + 3 + \ldots + 999$
    a) 9990
    b) 99900
    c) 999000
    d) 500000
    e) 499500
11. Assuming a uniform distribution the worst case and average complexities of Insertion sort are :
    a) $\sim n/2$ and $\sim n/4$
    b) $\sim n\log_2 n$ and $\sim \log_2 n/2$
    c) $\sim 4n$ and $\sim 2n$
    d) $\sim n^2/2$ and $\sim n^2/4$
    e) $\sim 4n^2/2$ and $\sim 2n^2$
12. The worst-case complexity of Euclid's algorithm for computing the gcd(a,b) occurs when a and b are :
    a) Relatively prime
    b) Different by one
    c) Consecutive prime numbers
    d) Consecutive Exponential numbers
    e) Consecutive Fibonacci numbers
13. The worst-case complexity of Quicksort occurs when :
    a) The list is already sorted
    b) The largest element occurs in the first position
    c) The input list satisfy a uniform distribution
    d) The input list has large size
    e) None of the above
14. Which of the following sorting algorithms is not in-place
    a) Bubble sort
    b) Insertion Sort
    c) Merge Sort
    d) Quicksort
    e) They are all in-place
15. The number of binary digits of an integer n is approximately equal to:
    a) 2^n
    b) $n \log_2 n$
    c) $\log_2 n$
    d) n^2
    e) None of the above
16. The encryption algorithm RSA employs the following algorithms:
    a) Powers and Horner's rule
    b) Powers and FFT
    c) Randomized Quicksort
    d) Max and Min
    e) Powers and Extended Euclid GCD

17. For which number of nodes does there exists a full binary tree:
    a) 16
    b) 25
    c) 63
    d) 100
    e) None of the above
18. The number of nodes of a 2-tree having 100 leaf nodes is:
    a) 99
    b) 100
    c) 101
    d) 199
    e) 200

Simple Answers

For each of the following give the **best** answer using simplest formula.

a) $1^{11} + 2^{11} + \ldots + n^{11} \sim$ ____________________________

b) $n + 2n^2 + \ldots + 10n^{10} \sim$ ____________________________

c) $1/2 + 1/3 + \ldots 1/(2n) \sim$ ____________________________

d) $\log n! \in \Theta($ $)$

e) Worst-case complexity of *BinarySearch* is $W(n) \sim$ ____________________________

f) lower bound for worst-case complexity of any comparison-based sorting algorithm is

$W(n) \in \Omega($ $)$

g) Average complexity of *MergeSort* is $A(n) \in \Theta($ $)$

h) Worst-case complexity of *QuickSort* is $W(n) \in \Theta($ $)$

i) Worst-case complexity of *Interpolation search* is $W(n) \in \Theta($ $)$

j) Average complexity of *Quicksort* with uniform distribution is

$A(n) \sim$ ____________________________

k) The number of decimal digits of a positive integer $n \sim$ ____________________________

l) The minimum depth of a binary tree is approximately ____________________________

More detailed Answers. Show Steps.

1. Show the action of the left-to-right binary method in computing x^{103}.

2. a) Give the recurrence relation for gcd(a,b) on which Euclid's algorithm is based.

   b) Show the action of Euclid's algorithm for a = 108 and b = 132.

.

3. Prove (using limits) that $O(n^k) \subset O(e^n)$, for any positive constant k.

4. Prove the average complexity $2n \ln n$ of *Quicksort* is of smaller order than the average complexity $n^2/4 - n/4$ of *Insertionsort*, i.e., show

$$O(2n \ln n) \subset O(n^2/4 - n/4).$$

5. Give pseudocode for the algorithm for evaluating a polynomial of degree n using *Horner's rule*, involving only n multiplications and n additions.

6. a) Define what is meant by the best-case complexity $B(n)$ of an algorithm.

   b) Define what is meant by the worst-case complexity $W(n)$ of an algorithm. . .

   .

   c) Define what is meant by the average complexity $A(n)$ of an algorithm.

7. Prove that the relation $f\ \Theta\ g$ whenever $f(n) \in g(n)$ is an *equivalence relation*.

8. Let $P(n)$ be a polynomial of degree k whose degree k term as coefficient a_k. Prove that

$$P(n) \sim a_k n^k$$

9. a) Derive a recurrence relation for the worst-case complexity $W(n)$ of *Mergesort*.

   .

   b) Solve the recurrence relation you have given in part b. SHOW STEPS.

10. Prove (using limits) that $O(\ln n) \subset O(n)$, for any positive constant k.

    .

11. Prove that H(n) ~ ln n, where H(n) = 1 + ½ + 1/3 + … + 1/n

12. Give a formula for the average complexity $A(n)$ involving the probabilities p_i, i = 1, 2, ..., where p_i is the probability that the algorithm performs i basic operations for an input of size n.

13. a) Give pseudocode for *Quicksort* (Do NOT include pseudocode for *Partition*).

    b) Analyze the worst-case behavior $W(n)$ of *Quicksort*.

14. For k a constant, prove that $\log n! = \log 1 + \log 2 + \ldots + \log n = \Theta(n \log n)$.

15. Define what is meant by *exponential* order.

16. Show that the number of decimal digits of n is approximately $\log_{10} n$.

17. Explain how both the maximum and minimum element in a list of size n (assume even) can be found in time $3n/2 - 2$

18. a) Consider a list $L[0: n - 1]$ of distinct elements. Compute $A(n)$ for *Linearsearch* assuming the search element X is in the list $L[0: n - 1]$ and equally likely to occur in any position.

    b) Repeat part a) assuming that X occurs on the list with probability .6 and is equally likely to occur in any position given that X is on the list.

19. Compute $A(n)$ for *Insertionsort* assuming a uniform distribution. You may assume that the average number of comparisons to insert the j^{th} element is $1/j$.

20. a) Obtain a recurrence relation for $A(n)$ for *Quicksort* assuming a uniform distribution.

    b) Solve the recurrence relation you have given in part a.

21. Prove that a binary tree having n nodes has depth $\Omega(\log n)$.

22. Discuss three ADTs for implementing a priority queue and compare them, i.e., discuss advantages and disadvantage.