THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2011
Campus: City

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: ONE hour)

NOTE: Attempt *all* questions!
Use of calculators is NOT permitted.

1. Which sorting algorithms covered in this course are comparison-based, stable, in-place and have asymptotically optimal running time?

   A. quicksort, heapsort
   B. heapsort, mergesort
   C. mergesort, quicksort
   D. none
   E. insertion sort, heapsort

2. Which of these is NOT in $\Omega(n \log n)$?

   A. $0.001n^2/\lg n$
   B. $17n \ln n$
   C. $n^{3/4}(\ln n)^5$
   D. $\sqrt{n^3 \log_{10} n}$
   E. $n^2(2 - \cos(n))$

3. A certain quadratic time algorithm uses 300 elementary operations to process an input of size 10. What is the most likely number of elementary operations it will use if given an input of size 1000?

   A. 3 000
   B. 30 000
   C. 3 000 000
   D. 300 000
   E. 300 000 000

4. Choose the strongest true statement: Selection sort makes at least as many comparisons as insertion sort

   A. for input with fewer inversions than average
   B. on every input
   C. in the worst case
   D. on reverse sorted input
   E. on average for uniformly random input

5. An algorithm whose running time for input size n satisfies the recurrence relation (for $n \geq 1$)

   $$T(n) = \frac{1}{n}\left(T(0) + T(1) + \ldots + T(n-1)\right) + 5n$$

   has running time in

   A. $\Theta(\log n)$
   B. $\Theta(n \log n)$
   C. $\Theta(\sqrt{n})$
   D. $\Theta(n^2)$
   E. $\Theta(n)$

6. You're trying to break someone's sorting implementation but can't see their source code. You notice that the algorithm runs in a similar time for both ordered and unordered datasets, and performance is good on large ones. Items with the same key will sometimes have their relative order changed. Which sorting algorithm are they most likely using?

A. quicksort
B. selection sort
C. heapsort
D. insertion sort
E. mergesort

7. The running time for the following code fragment is $\Theta(f(n))$. What is $f(n)$?

```
for(int i = 1; i < n; i = i * 2){
    for(int j = 0; j < n; j++){
        if(i == j){
            for(int k = 0; k < n; k++){
                // Do something elementary
            }
        }
        else{
            // Do another elementary thing
        }
    }
}
```

A. $n^2 \log n$
B. $n \log n$
C. n
D. $n(\log n)^2$
E. n^2

8. The minimum possible number of comparisons made in the worst case by a comparison-based sorting algorithm in order to sort a list of size 4 is

A. 5
B. 4
C. 6
D. 8
E. 3

9. You know that algorithm A runs in exponential time $\Theta(2^n)$. If your computer can process input of size 1000 in one year using an implementation of this algorithm, about what size input would you expect to be able to solve in one year with a computer 1000 times faster?

   A. 32 000
   B. 1 010
   C. 1 500
   D. 1 002
   E. 1 000 000

10. The recurrence $T(n) = T(n/2) + n^2, T(1) = 0$ has a solution that is of exact order

    A. n
    B. n^2
    C. 2^n
    D. $n \log n$
    E. $\log n$

11. After the items 4, 6, 1, 3, 5, 2 are inserted in that order into an initially empty binary max-heap, what is the right child of the root?

    A. 1
    B. 4
    C. 5
    D. 2
    E. 3

12. Which of the following inputs will make the standard implementation of quicksort (always choose the leftmost element as the pivot) do the least number of comparisons?

    A. 4,3,1,2,6,5,7
    B. 7,6,5,4,3,2,1
    C. 1,2,3,4,5,6,7
    D. 1,3,5,7,6,4,2
    E. 2,4,6,7,5,3,1

13. If algorithm A has quadratic time complexity and runs for 1 second on a problem of size 100, about how long would you expect it to take to solve a problem of size one million?

    A. 3 years
    B. 300 years
    C. 12 days
    D. 4 months
    E. 3 hours

14. Choose the FALSE statement: $7\lg n + 5n + n\lg\lg n + 3n\ln n$ is

A. $\Omega(n)$
B. $\Theta(n\log n)$
C. $\Omega(n^2)$
D. $O(n^2)$
E. $O(n\log^2 n)$

15. For a tree, the *** is defined to be the maximum of all depths of nodes. What is ***?

A. total depth
B. width
C. order
D. internal path length
E. height

16. A certain cubic time algorithm uses 30 elementary operations to process an input of size 10. What is the most likely number of elementary operations it will use if given an input of size 1000?

A. 3 000
B. 30 000
C. 300 000
D. 3 000 000
E. 30 000 000

17. The recurrence $T(n) = T(n/2) + n^2, T(1) = 0$ has exact solution (for n a power of 2):

A. $2n - 2$
B. $(4n^2 - 4)/3$
C. $\lg n$
D. $(4^n - 4)/3$
E. $2^n - 2$

18. Consider the following statements. Which ones are true?

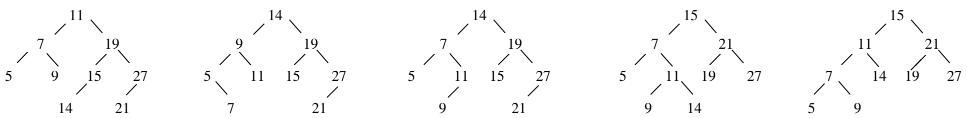
(i) If $f(n) = n^2, g(n) = (1 + (-1)^n)n$, then $g(n)$ is $O(f(n))$
(ii) If $f(n) = n, g(n) = (1 + (-1)^n)n^2$, then $g(n)$ is $\Omega(f(n))$
(iii) If $f(n) = n^2\log_4(n), g(n) = (5n^2 + 1)(\lg n + \lg\lg n)$, then $f(n)$ is $\Theta(g(n))$

A. i, ii
B. ii, iii
C. none
D. all
E. i, iii

19. You are presented with an unknown sorting algorithm ***-sort. The algorithm appears to run at a constant speed on both random and sorted data sets, and does not markedly slow down as they increase in size. Any two items which have the same search key have their relative order unchanged. Which of the following is *** most likely to be?

A. selection
B. insertion
C. quick
D. merge
E. heap

20. How many of the 5 binary trees shown are complete?



One Two Three Four Five

A. 1
B. 5
C. 2
D. 3
E. 0

Working page 1

Working page 2