THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2012
Campus: City

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: 50 minutes)

NOTE:

- Enter your name and student ID into the Teleform sheet FIRST.
- THEN: Attempt *all* 20 questions!
- All questions have ONE correct answer.
- DO NOT tick two answers as correct for the same question.
- If you believe that there is an error in a question (multiple correct answers or no correct answer), select the answer you believe was intended as the correct one and contact us after the test.
- Keep your question book. Writing on the question book will not be marked.
- Use of calculators is NOT permitted.
- Good luck!

1. You know that algorithm A runs in exponential time $\Theta(2^n)$. If your computer can process input of size 100 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. 110
B. 1 00 000
C. 1024
D. 10000
E. 1500

2. Suppose that the function $f(n)$ is $\Theta(n)$ and that the function $g(n)$ is $O(n \log n)$. Then the product function $h(n) = f(n) \cdot g(n)$ is ...

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

3. Which of the following inputs makes selection sort perform the largest number of comparisons?

A. 2, 3, 5, 4, 1
B. 3, 5, 4, 1, 2
C. 1, 2, 3, 4, 5
D. all the other choices are correct
E. 5, 4, 3, 2, 1

4. 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. heap
C. merge
D. insertion
E. quick

5. The *stupid sort* algorithm works as follows to sort a list in place: Starting from start of the list, scan until two succeeding items that are in the wrong order are found. Swap those items and go back to the beginning. The algorithm ends when the end of the list is reached.

   The worst-case running time for a list of size n is of order:

   A. $n!$
   B. n^2
   C. $n \log n$
   D. n^3
   E. n^n

6. Which of the following bit strings causes a Lempel-Ziv parser to find a known pattern that overlaps with the start of pattern's repetition from the current parsing position?

   A. 0100110 . . .
   B. 1011010 . . .
   C. 0110100 . . .
   D. 0101011 . . .
   E. 0011000 . . .

7. You write a program in which the user inputs names one at a time, with an empty name signaling the end of input. Which of the following data structures is best suited to store such a list of names and output in reverse?

   A. binary tree
   B. stack
   C. priority queue
   D. heap
   E. queue

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

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

9. If algorithm A has quadratic time complexity and runs for 1 second on a problem of size 200, about how long would you expect it to take to solve a problem of size two million?

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

10. The array [9,5,6,3,2,15] represents a complete binary tree in the usual way. Make it into a heap by percolating up or down a single element. What is the resulting array?

A. [3,9,6,5,2,15]
B. [15,5,6,3,2,9]
C. [6,5,9,3,2,15]
D. [9,5,15,3,2,6]
E. [15,5,9,3,2,6]

11. Given a sorted array of 11 integers $[5\ 9\ 13\ 17\ 22\ 29\ 35\ 41\ 47\ 51\ 55]$ you are asked to use Binary Search to search for the value 41. The search's low and high pointers are initially set to 0 and 10. What are their values after *two* cycles (i.e. halvings of the search size)?

A. low=5, high=7
B. low=6, high=7
C. low=6, high=8
D. low=7, high=7
E. low=5, high=8

12. 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. insertion sort
D. mergesort
E. heapsort

13. Consider an LZ77 parser with a window size (for both the look-behind and look-ahead window) of 4 bits operating on a binary string. Assuming that the parser terminates the search in a step as soon as a maximum length match is found, what is the largest number of bit comparisons performed in a step?

A. 16
B. $\Theta(n)$
C. 10
D. 24
E. 13

14. After the items 8, 12, 2, 6, 10, 3 are inserted in that order into an initially empty binary max-heap, what is the right child of the root?

A. 2
B. 8
C. 6
D. 10
E. 3

15. Which of the following inputs makes mergesort perform the largest number of comparisons?

A. 2, 1, 4, 3
B. 1, 2, 3, 4
C. 4, 3, 2, 1
D. 1, 3, 2, 4
E. all the other choices are correct

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

17. 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. 3 000 000
B. 30 000 000
C. 30 000
D. 300 000
E. 3 000

18. Consider all possible comparison-based sorting algorithms applied to sort a list of size 4. For each algorithm a, let $W(a)$ be the worst case number of comparisons to sort such a list. The minimum possible value of $W(a)$ is:

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

19. Let the function $f(n)$ be $\Theta(n^2)$. Then the function $f(n)$ is simultaneously ...

A. $O(n)$ and $\Omega(n^3)$
B. $O(n^3)$ and $\Omega(n^3)$
C. $O(n^3)$ and $\Omega(n^2)$
D. $O(n^2)$ and $\Omega(n^3)$
E. $O(n)$ and $\Omega(n^2)$

20. 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
B. $n(\log n)^2$
C. $n^2 \log n$
D. n
E. $n \log n$

Working page 1

Working page 2