THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2014
Campus: City

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: 40 minutes)

NOTE:

- Enter your name and student ID into the Teleform sheet FIRST.
- THEN: Attempt *all* 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 the test room supervisor 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. Which of the following statements about heapsort is FALSE?

   A. To add a node to a heap, one adds it into the first free position, then "bubbles it up" to its correct position
   B. To delete a heap node, one replaces it with the heap's last child node, then "percolates it down" to its correct position
   C. Although heapsort is a recursive algorithm, it can be implemented without using recursion
   D. Heapsort is stable
   E. Best, worst and average performance of heapsort is $O(n \log n)$

2. Which of the following statements is FALSE?

   A. Insert/delete operations are $O(1)$ for ordered lists
   B. It's slow to search a list for a particular element
   C. Arrays allow $O(1)$ random read/write operations
   D. Average search time for a list is $\Theta(n)$
   E. It's slow to insert a new element into an array

3. Which of the following statements about hash collisions is FALSE?

   A. Collision probability depends on the number of elements in the array
   B. Collisions can be handled by linking elements with the same hash code into 'hash chains'
   C. Collisions can be handled using an "open addressing" algorithm
   D. Collisions don't occur if we use a perfect hash function
   E. Collisions are unlikely to occur very often

4. Assume we have 2 algorithms, A and B, which use exactly $c_A * n^2$ and $c_B * n$ primitive operations. $T_A(10) = 10$ and $T_B(10) = 100$, for what n will A be faster?

   A. $n > 10$
   B. $n < 10$
   C. $n < 100$
   D. none of these
   E. $n > 100$

5. Given the following algorithm description:

```
def p(n) (
    for i from 1 to n do
        r(n)
    end
)
```

where r(n) has complexity of $\Theta(n^2)$, what is the resulting runtime complexity of p(n)?

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

6. An Operating System must provide 'heap' memory so that programs can use arbitrary chunks of memory on demand. For which of the following is the OS's memory manager NOT responsible?

A. Concatenate adjacent free chunks into bigger single chunks
B. Check for program attempts to access memory not allocated to that program
C. Respond to requests for memory chunks of a specified size
D. Put memory chunks returned by programs back into its free list
E. Maintain a list of free memory chunks

7. A certain algorithm processes a list of size n by first inspecting the first and last elements of the list, splitting the list into 2 sublists of equal size, then recursively processing the sublists. It gives rise to a recurrence for the running time $T(n)$ that looks like $T(n) =$***. What is ***?

A. $2T(\frac{n}{2}) + 2$
B. $2T(\frac{n}{2}) + n$
C. $T(\frac{n}{2})$
D. $2T(\frac{n}{2}) + 1$
E. $T(\frac{n}{2}) + n$

8. Which of the following input $\rightarrow$ output sequences, where the elements whose names start with the same letter compare as equal (e.g. $A_1 == A_2$), demonstrate a *stable* sort algorithm?

   A. $A, D, B_1, C, B_2 \rightarrow A, B_2, C, B_1, D$
   B. $A, D, B_1, C, B_2 \rightarrow A, B_1, B_2, C, D$
   C. $A, D, C, E, B \rightarrow A, B, C, D, E$
   D. $E, C_1, A, C_2, C_3 \rightarrow A, C_3, C_2, C_1, E$
   E. $A_1, A_2, C, A_3, A_4 \rightarrow A_1, A_4, A_2, A_3, C$

9. Consider an array a with n elements. For a section of the array from index low to index $high$, let $middle = \lfloor (low + high)/2 \rfloor$
   Which of the following is a *good* choice for a pivot value if you want to sort the section using quicksort?

   A. $median(a[low], a[middle], a[high])$
   B. $a[0]$
   C. $a[middle]$
   D. $a[n-1]$
   E. $a[r]$, where r is a random integer in the range 0 to $n-1$

10. If $f(n)$ is $O(n^3)$ and $g(n)$ is $\Omega(n)$ which of the following is true about the function $h(n) = f(n) * g(n)$?

    A. $O(n^3)$
    B. $\Theta(n^3)$
    C. none of these
    D. $\Omega(n^3)$
    E. $\Theta(n^4)$

11. If $f(n)$ is $\Omega(n^3)$ and $g(n)$ is $\Omega(n)$ which of the following is FALSE about the function $h(n) = f(n) + g(n)$?

    A. $h(n)$ is $\Omega(n^3)$
    B. $h(n)$ is $\Omega(n^4)$
    C. $h(n)$ is $\Omega(n^2)$
    D. $h(n)$ is $\Omega(n \log n)$
    E. $h(n)$ is $\Omega(n)$

12. You test an algorithm that is known to exhibit quadratic time complexity. With an input size of n = 2, the algorithm runs for a day. As you double the input size (n = 4), the algorithm takes 4 days to complete. How long is input size 10 likely to take?

    A. 100 days
    B. 25 days
    C. 10 days
    D. 256 days
    E. 16 days

13. Given

```
def p(n) =
    (
        if n > 1
            f(n);
            p(n-1);
            p(n-1);
        else
            f(n)
    )
```

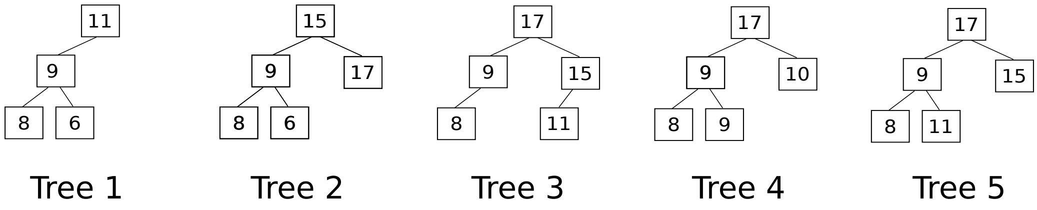
Assume that the runtime for f(n) is constant, i.e., it always takes exactly the same amount of time, regardless of n. Assume also that p(1) takes exactly one second to run. How long does it take p(7) to run?

    A. 71 seconds
    B. 127 seconds
    C. 64 seconds
    D. none of these
    E. 7 seconds

14. Consider hash functions that return a hash code which is used to index a hash table with m elements. Which of the following hash function properties is the LEAST important?

    A. Its value can be computed in $O(1)$ operations
    B. It always returns an integer value in $[0..m-1]$
    C. It uses only arithmetic operations on parts of the key and numeric constant
    D. It produces the same hash code for identical keys
    E. It distributes its hash codes uniformly across $[0..m-1]$

15. Which of the following trees is a maximum heap?



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

16. An algorithm whose running time for input size n satisfies the recurrence relation (for $n \geq 1$) $T(n) = 3 * T(\frac{n}{3}) + 2 * n$ has running time in

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

17. Choose the FALSE statement: $n * (3n + \log n) + 5n + \frac{7*n^2}{n^3}$ is

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

18. Which of the following statements about m-ary search trees is FALSE?

A. If all the nodes are stored on disk, they reduce the total I/O time for searches
B. They have up to m keys in each node
C. They store all their data items in their leaf nodes
D. They require up to $m - 1$ comparisons at each node
E. They reduce the depth of the tree, compared to a binary tree

Working page 1

Working page 2