Exam questions

Mark Allen Weiss
School of Computer Science
Florida International University
University Park
Miami, FL 33199

Abstract

This report contains a collection of multiple-choice questions, organized by book chapter that can be used for examinations.[1] Answers are provided at the end. This report was typeset in LaTeX. Original source is available.

Chapter 1

1. What is the approximate value of log 1,000,000?
   (a) 10
   (b) 20
   (c) 50
   (d) 1000
   (e) none of the above

2. If $\log n$ equals 100, what is the value of $\log(2n)$?
   (a) 101
   (b) 200
   (c) 1000
   (d) 10000
   (e) none of the above

3. When performing a proof by induction, which is the case that is trivially true?
   (a) the basis
   (b) the inductive hypothesis
   (c) the lemma
   (d) the theorem
   (e) none of the above

[1] Many of these questions appear in different form in the *Instructor's Resource Manual* for *Algorithms, Data Structures, and Problem Solving with C++*, published by Addison-Wesley, 1996.

4. The following routine violates which rule(s) of recursion?

```
function Recurse( N : Integer ) return Integer is
begin
    if N = 0 then
        return 0;
    else
        return N + Recurse( N / 2 ) + Recurse( N / 2 + 1 );
    end if;
end Recurse;
```

   (a) No base case
   (b) Fails to make progress
   (c) Performs redundant work
   (d) Two of the above
   (e) All of (a), (b), and (c)

5. Which of the following is the most likely result of failing to make progress towards a base case in a recursive call?

   (a) compiler enters into an infinite loop
   (b) error at compilation time
   (c) exception is raised at run time
   (d) recursive routine enters an infinite loop when it runs
   (e) recursive routine terminates with bad value but no other error

   Answers: 1-B, 2-A, 3-A, 4-D, 5-C.

Chapter 2

1. Which of the following functions grows fastest?

   (a) $n \log n$
   (b) 2^n
   (c) $\log n$
   (d) n^2
   (e) n^{20}

2. Which of the following functions grows fastest?

   (a) $n + \log n$
   (b) $n \log n$
   (c) $n - \log n$
   (d) n
   (e) There is a tie among two or more functions for fastest growth rate

The next three questions apply to the following code fragment:

```
1     for i in 1..N loop
2         for j in 1..i loop
3             for k in i..j loop
4                 Sum := Sum + 1;
5             end loop;
6         end loop;
7     end loop;
8     for p in 1..N*N loop
9         for q in 1..p loop
10            Sum := Sum - 1;
11        end loop;
12    end loop;
```

3. How many times is statement 4 executed?

   (a) $O(N)$
   (b) $O(N^2)$
   (c) $O(N^3)$
   (d) $O(N^4)$
   (e) none of the above

4. How many times is statement 10 executed?

   (a) $O(N)$
   (b) $O(N^2)$
   (c) $O(N^3)$
   (d) $O(N^4)$
   (e) none of the above

5. What is the running time of the fragment?

   (a) $O(N^4)$
   (b) $O(N^5)$
   (c) $O(N^6)$
   (d) $O(N^7)$
   (e) none of the above

6. Suppose $T_1(n) = O(F(n))$ and $T_2(n) = O(F(n))$. Which of the following are true?

   (a) $T_1(n) + T_2(n) = O(F(n))$
   (b) $T_1(n) * T_2(n) = O(F(n))$
   (c) $T_1(n)/T_2(n) = O(1)$
   (d) $T_1(n) = O(T_2(n))$
   (e) none of the above

7. Programs A and B are analyzed and found to have worst-case running times no greater than $150n \log n$ and n^2, respectively. Which of the following statements does the analysis imply?
   (a) Program A will run faster on average for sufficiently large n.
   (b) Program B will run faster on average for small n.
   (c) Program A is probably simpler to code than program B.
   (d) There exists some input for which program B takes longer than program A.
   (e) none of the above

8. An algorithm takes 10 seconds for an input size of 50. If the algorithm is quadratic, approximately how long does it take to solve a problem of size 100?
   (a) 10 seconds
   (b) 20 seconds
   (c) 40 seconds
   (d) 100 seconds
   (e) none of the above

9. An algorithm takes 30 seconds for an input of size 1000. If the algorithm is quadratic, how large a problem can be solved in two minutes?
   (a) 2000
   (b) 4000
   (c) 6000
   (d) 60000
   (e) none of the above

10. An algorithm takes 6 seconds to solve a problem of size 100 and ten minutes to solve a problem of size 1000. What is the likely running time of the algorithm?
    (a) constant
    (b) linear
    (c) quadratic
    (d) cubic
    (e) none of the above

11. Which of (a) to (d) is false about the binary search?
    (a) the input array must be sorted
    (b) successful searches take logarithmic time on average
    (c) unsuccessful searches take logarithmic time on average
    (d) the worst case for any search is logarithmic
    (e) all of the above are true

12. Which of the following can be done in $O(\log n)$ arithmetic operations?

(a) Raising a number to the nth power

(b) Computing the greatest common divisor of some integer and n

(c) Adding two n-digit numbers

(d) Two of the above

(e) All of (a), (b), and (c)

13. A recursive algorithm works by solving two half-sized problems recursively, with an additional linear-time overhead. The total running time is most accurately given by

    (a) $O(\log n)$

    (b) $O(n)$

    (c) $O(n \log n)$

    (d) $O(n^2)$

    (e) none of the above

14. The solution to $T(n) = T(\lfloor 3n/4 \rfloor) + 10$ with $T(0) = 0$ is most accurately given by

    (a) $O(\log n)$

    (b) $O(n)$

    (c) $O(n \log n)$

    (d) $O(n^2)$

    (e) none of the above

15. Approximately how many random numbers are using in the permutation generation algorithm in Exercise 2.7.c?

    (a) 1

    (b) $\log n$

    (c) n

    (d) $n \log n$

    (e) none of the above

16. What is the running time of the following routine?

```
// Check if N is prime
function Is_Prime( N : Integer ) return Boolean is
    I : Integer := 3;
begin
    if N = 2 or else N = 3 then
        return TRUE;
    end if;
    if N MOD 2 = 0 then
        return FALSE;
    end if;
    while i * i <= N loop
        if N MOD i = 0 then
```

```
            return FALSE;
        else
            I := I + 2;
        end if;
    end loop;
    return TRUE;
end Is_Prime;
```

(a) constant time
(b) $O(\log N)$
(c) $O(N)$
(d) $O(\sqrt{N})$
(e) none of the above

Answers: 1-B, 2-B, 3-C, 4-D, 5-A, 6-A, 7-E, 8-C, 9-A, 10-C, 11-E, 12-D, 13-C, 14-A, 15-C, 16-D.

Chapter 3

1. Which of the following operations is not efficiently supported by a singly-linked list?

   (a) accessing the element in the current position
   (b) insertion after the current position
   (c) insertion before the current position
   (d) moving to the position immediately following the current position
   (e) all of the above are efficiently supported

2. Which statement, placed in the list package implementation, inserts an item `X` after position `Current`?

   (a) `Current := new Node'( X, Current );`
   (b) `Current := new Node'( X, Current.Next );`
   (c) `Current.Next := new Node'( X, Current );`
   (d) `Current.Next := new Node'( X, Current.Next );`
   (e) none of the above

3. The header node of a linked list

   (a) simplifies deletion
   (b) simplifies insertion
   (c) uses only constant extra space
   (d) two of the above
   (e) all three of (a), (b), and (c)

4. If a header node is used, which of the following indicates a list `L` with one item?

   (a) `L.Header.Next = null`

(b) `L.Header.Next /= null`

(c) `L.Header.Next /= null and then L.Header.Next.Next /= null`

(d) `L.Header.Next /= null and then L.Header.Next.Next = null`

(e) none of the above

5. Insertion of a node into a doubly linked list requires how many changes to various `Next` and `Prev` pointers?

   (a) no changes

   (b) 1 `Next`, 1 `Prev`

   (c) 2 `Next`, 2 `Prev`

   (d) 3 `Next`, 3 `Prev`

   (e) none of the above

6. What operation is supported in constant time by the doubly linked list, but not by the singly linked list?

   (a) `Advance`

   (b) `Backup`

   (c) `First`

   (d) `Retrieve`

   (e) all of the above are always constant time

7. The UNIX editor *vi* allows searching in both directions, with wraparound if necessary. If the sequence of lines is stored as a linked list, which of the following is most reasonable?

   (a) singly linked list

   (b) doubly linked list

   (c) circular singly linked list

   (d) circular doubly linked list

   (e) none of the above

8. What happens when wraparound is implemented for a queue?

   (a) If `Front` advances past the last array position, it is reset to the first array position.

   (b) If `Rear` advances past the last array position, it is reset to the first array position.

   (c) Both (a) and (b)

   (d) Neither (a) nor (b)

9. Using the text implementation, if `Front` and `Rear` have identical values, what is the size of the queue?

   (a) 0

   (b) 1

   (c) 2

   (d) the answer cannot be determined

(e) none of the above

10. For the linked list implementation of the stack, where are the pushes and pops performed?

    (a) Push in front of the first element, pop the first element
    (b) Push after the last element, pop the last element
    (c) Push after the last element, pop the first element
    (d) Push in front of the first element, pop the last element
    (e) Push after the first element, pop the first element

11. For the linked list implementation of the queue, where are the enqueue and dequeues performed?

    (a) Enqueue in front of the first element, dequeue the first element
    (b) Enqueue after the last element, dequeue the last element
    (c) Enqueue after the last element, dequeue the first element
    (d) Enqueue in front of the first element, dequeue the last element
    (e) Enqueue after the first element, dequeue the first element

12. For the linked list implementation, if the stack is not empty, which of the following statements in a main procedure can be used to access the top element in the stack `S`?

    (a) `S.Element`
    (b) `S.TopOfStack`
    (c) `S.TopOfStack.Element`
    (d) `TopOfStack.Element`
    (e) none of the above

Answers: 1-C, 2-D, 3-E, 4-D, 5-C, 6-B, 7-D, 8-C, 9-B, 10-A, 11-C, 12-E.

Chapter 4

1. Which of the following traversals requires more than linear time in the worst case?

    (a) inorder
    (b) level order
    (c) postorder
    (d) preorder
    (e) all of these traversals are linear time

2. In which of the following traversals is the node processed before the recursive calls to the children complete?

    (a) inorder
    (b) level order
    (c) postorder

(d) preorder
(e) none of the above

3. What is the maximum number of nodes in a binary tree with L leaves?

    (a) $2L$
    (b) 2^L
    (c) 2^{L+1}
    (d) there is no maximum
    (e) none of the above

4. Which of the following is true about the height of a node?

    (a) The height of a node is one less than the height of its parent
    (b) The height of an empty tree is 0
    (c) The height of a leaf is 0
    (d) The height of a tree can be larger than its depth
    (e) all of the above are false

5. The first child / next sibling implementation

    (a) allows easy access to the parent
    (b) is appropriate for binary trees
    (c) uses C pointers per node, where C is the number of children
    (d) all of the above
    (e) none of (a), (b), and (c)

6. Which traversal computes the total size of each directory in the UNIX file system?

    (a) inorder
    (b) level order
    (c) postorder
    (d) preorder
    (e) two or more of the above traversals could be used

7. Let `C(X)` be the number of leaves in a binary tree rooted at `T`. Assume that `IsLeaf(T)` returns 1 if `T` is a leaf. Which of the following observations leads to a recursive implementation?

    (a) `C(T):=C(T.Left)+C(T.Right)`
    (b) `C(T):=C(T.Left)+C(T.Right)+1`
    (c) `C(T):=C(T.Left)+C(T.Right)+IsLeaf(T)`
    (d) `C(T):=C(T.Left)+C(T.Right)+IsLeaf(T)+1`
    (e) none of the above

8. Which traversal does not use a stack?

(a) inorder
(b) level order
(c) postorder
(d) preorder
(e) all of these traversals uses a stack

9. How many n node binary trees with items 1, 2, ..., n have identical postorder and inorder traversals?

   (a) 0
   (b) 1
   (c) n
   (d) $n!$
   (e) none of the above

The next three questions relate to the binary tree with root A. The root has left child B and right child C. B has left child D and right child E. There are no other nodes in the tree.

10. Which of the following traversals yields ABCDE?

    (a) inorder
    (b) level order
    (c) postorder
    (d) preorder
    (e) two of the above

11. Which of the following is an inorder traversal of the tree?

    (a) ABCDE
    (b) ABDEC
    (c) DBEAC
    (d) DEBAC
    (e) none of the above

12. The height of the tree is

    (a) 0
    (b) 1
    (c) 2
    (d) 3
    (e) none of the above

13. Approximately what is the maximum height of a binary search tree of n nodes?

    (a) $\log n$
    (b) $1.38 \log n$

(c) $1.44 \log n$
(d) $2 \log n$
(e) none of the above

14. The following items are inserted into a binary search tree: 8, 3, 4, 9, 5, 6, 2, 1, 7. Which item is placed at a root?

    (a) 1
    (b) 4
    (c) 8
    (d) 9
    (e) none of the above

15. The following items are inserted into a binary search tree: 3, 6, 5, 2, 4, 7, 1. Which node is the deepest?

    (a) 1
    (b) 3
    (c) 4
    (d) 7
    (e) none of the above

16. Which of the following statements is true about deleting the root of a binary search tree?

    (a) the root pointer always changes
    (b) the root pointer changes if it does not have two children
    (c) if the root has two children, its item is replaced by the largest element in the right subtree
    (d) all of the above
    (e) none of (a), (b), and (c)

17. For an insertion of a single item into an n-item AVL tree, the maximum number of rotations (double rotations count as one rotation) is

    (a) 1
    (b) 2
    (c) approximately $\log n$
    (d) approximately $1.44 \log n$
    (e) none of the above

18. The following items are inserted into an AVL tree: 1, 2, 3, 8, 6. How many rotations are performed?

    (a) no rotations
    (b) 1 single rotation only
    (c) 1 double rotation only

(d) 1 single rotation and 1 double rotation

(e) none of the above

19. Items 7, 3, 11, 9, and 13 are inserted into an AVL tree. What happens when 12 is inserted?

    (a) no rotation is needed

    (b) a single rotation between some node and its left child is performed

    (c) a single rotation between some node and its right child is performed

    (d) a double rotation with a node, its left child, and a third node is performed

    (e) a double rotation with a node, its right child, and a third node is performed

20. Which of the following data structures has the strongest height guarantee?

    (a) AVL tree

    (b) B-tree of order 3

    (c) B-tree of order 5

    (d) splay tree

    (e) 2-3 tree

21. Suppose a disk block stores 4096 bytes and the basic key size is 96 bytes. Assuming that pointers cost 4 bytes, what is the correct choice of M for a B-tree?

    (a) 41

    (b) 42

    (c) 43

    (d) 96

    (e) none of the above

22. In addition to the data and left and right pointers, for **any** implementation, what must be stored in each node of a splay tree?

    (a) the node's height

    (b) the node's level

    (c) the node's parent

    (d) the node's rank

    (e) none of the above

23. What is amortized cost of an operation using rotate-to-root?

    (a) $O(1)$

    (b) $O(\log n)$

    (c) $O(n)$

    (d) $O(n \log n)$

    (e) none of the above

24. An access of a splay tree of n nodes results in a completely identical tree. For how many different nodes would this be possible?

   (a) 0
   (b) 1
   (c) 2
   (d) $n-1$
   (e) none of the above

25. What is the worst-case height of a splay tree?

   (a) $\log n$
   (b) $1.38 \log n$
   (c) $2 \log n$
   (d) $n-1$
   (e) n

26. Which of the statements (a) to (d) about splay trees is false?

   (a) a single access operation could examine every node in the tree
   (b) any n consecutive operations from an initially empty splay tree must take at most $O(n \log n)$ time
   (c) inserting the items 1, 2, ..., n into an initially empty splay tree takes $O(n)$ total time.
   (d) the most recently accessed item is at the root
   (e) none of (a) to (d) is false

27. Which of the following splay tree rotations in effect distinguishes it from rotate-to-root?

   (a) zig only
   (b) zig-zag only
   (c) zig-zig only
   (d) zig-zig and zig-zag only
   (e) all of the above

28. What item is at the root after the following sequence of insertions into an empty splay tree: 1, 11, 3, 10, 8, 4, 6, 5, 7, 9, 2?

   (a) 1
   (b) 2
   (c) 4
   (d) 8
   (e) none of the above

29. How is deletion performed in a splay tree?

   (a) If the node is found, it is replaced with the smallest node in its right subtree, which itself is recursively deleted.

(b) It the node is found, it is replaced with the largest node in its left subtree, which itself is recursively deleted.
(c) A single splay is performed which places the deleted node in a leaf; that node is then easily removed
(d) A single splay is performed which places the deleted node at the root; it is deleted and the two subtrees are reattached by using a second splay
(e) none of the above

30. In a splay tree, how is the rank of a node stored?
    (a) an extra array stores the information
    (b) a linked list stores the information
    (c) directly, in each node
    (d) indirectly, by storing the size in each node
    (e) the rank is not stored at all

31. Which of the following alternatives preserves the logarithmic amortized time bound for the splay tree?
    (a) do not splay on unsuccessful searches
    (b) do not splay if an access path has fewer than $\log n$ nodes
    (c) replace the zig-zig with two single bottom-up rotations
    (d) splay on every other access
    (e) none of the above

Answers: 1-E, 2-D, 3-D, 4-C, 5-E, 6-C, 7-C, 8-B, 9-B, 10-B, 11-C, 12-C, 13-E, 14-C, 15-C, 16-B, 17-A, 18-D, 19-C, 20-C, 21-A, 22-E, 23-C, 24-B, 25-D, 26-E, 27-C, 28-B, 29-D, 30-E, 31-B.

Chapter 5

1. Which of the following data structures requires more than constant average time for insertions?
    (a) hash table
    (b) queue
    (c) search tree
    (d) stack
    (e) all of the above have constant time insertion algorithms

2. What is the range of values computed by the hash function $Hash(X) = X \bmod 100$?
    (a) 0 to 99
    (b) 0 to 100
    (c) 1 to 99
    (d) 1 to 100

(e) none of the above

3. Which of (a) to (d) is false: The size of a hash table
   (a) should be a power of 2 for quadratic probing
   (b) should be a prime number for linear probing
   (c) should be about $2n$ for quadratic probing
   (d) should be about n for separate chaining
   (e) two or more of the above are false

4. How are elements deleted in linear probing?
   (a) deletion is not allowed
   (b) they are changed to zero
   (c) they are marked deleted
   (d) unchecked deallocation
   (e) none of the above

5. Suppose we are implementing quadratic probing with a hash function $Hash(X) = X \bmod 100$. If an element with key 4594 is inserted and the first three locations attempted are already occupied, then the next cell that will be tried is
   (a) 2
   (b) 3
   (c) 9
   (d) 97
   (e) none of the above

6. In a separate chaining hash table with load factor $\lambda = 0.8$, what is the average length of a list?
   (a) 0.8
   (b) 1.0
   (c) 1.25
   (d) there is not enough information
   (e) there is enough information, but none of the above are correct

7. Which of the following costs are equal in a probing hash table?
   (a) insertion and successful search
   (b) insertion and unsuccessful search
   (c) successful search and unsuccessful search
   (d) insertion, successful search, and unsuccessful search
   (e) none of the above

8. Which of the following statements about quadratic probing is true (expensive does not include trivial operations such as multiplication or division by powers of 2; computation of the hash function is not included in the cost)?
   (a) an expensive division must be performed
   (b) an expensive mod operator must be performed
   (c) an expensive multiplication must be performed
   (d) all of the above
   (e) none of (a), (b), and (c)

9. Linked lists are used in
   (a) double hashing
   (b) linear probing
   (c) quadratic probing
   (d) separate chaining
   (e) all of the above

10. *Primary clustering* occurs in
    (a) linear probing
    (b) quadratic probing
    (c) separate chaining
    (d) all of the above
    (e) none of (a), (b), and (c)

11. *Rehashing* can be used in
    (a) linear probing
    (b) quadratic probing
    (c) separate chaining
    (d) all of the above
    (e) none of (a), (b), and (c)

Answers: 1-C, 2-A, 3-A, 4-C, 5-B, 6-A, 7-B, 8-E, 9-D, 10-A, 11-D.

Chapter 6

1. Every node in a (min) binary heap
   (a) has two children
   (b) is no larger than its children
   (c) is no smaller than its children
   (d) has a smaller left child than right child
   (e) two or more of the above

2. If an element in a binary heap is stored in position i and the root is at position 1, then where is the parent stored?

   (a) $\lfloor i/2 \rfloor$
   (b) $\lceil i/2 \rceil$
   (c) $1 + \lfloor i/2 \rfloor$
   (d) $2i$
   (e) $2i + 1$

3. The running time of `Build_Heap` is

   (a) $O(n)$ worst case and $O(n)$ average case
   (b) $O(n)$ worst case and $O(\log n)$ average case
   (c) $O(n)$ worst case and $O(n \log n)$ average case
   (d) $O(n \log n)$ worst case and $O(n)$ average case
   (e) $O(n \log n)$ worst case and $O(n \log n)$ average case

4. n elements are inserted one by one into an initially empty binary heap. The total running time is

   (a) $O(n)$ worst case and $O(n)$ average case
   (b) $O(n)$ worst case and $O(\log n)$ average case
   (c) $O(n)$ worst case and $O(n \log n)$ average case
   (d) $O(n \log n)$ worst case and $O(n)$ average case
   (e) $O(n \log n)$ worst case and $O(n \log n)$ average case

5. Which operation is not supported in constant time by a double-ended queue (deque)?

   (a) Insertion as the front or rear item
   (b) Access of the front or rear item
   (c) Deletion of the front or rear item
   (d) Access and deletion of the minimum item
   (e) all of the above are supported

6. Which operation is not efficiently supported by priority queues?

   (a) `Delete_Min`
   (b) `Find`
   (c) `Find_Min`
   (d) `Insert`
   (e) All of the above are efficiently supported

7. Which data structure is used to check for balanced parentheses?

   (a) binary search tree
   (b) hash table

(c) priority queue
(d) queue
(e) stack

8. Jobs sent to a printer are generally placed on a

    (a) binary search tree
    (b) hash table
    (c) priority queue
    (d) queue
    (e) stack

9. Which data structure is generally used to implement a symbol table?

    (a) binary search tree
    (b) hash table
    (c) priority queue
    (d) queue
    (e) stack

10. Which data structure maintains the event set in an event-driven (discrete-event) simulation?

    (a) binary search tree
    (b) hash table
    (c) priority queue
    (d) queue
    (e) stack

11. Which of the following could be used as an efficient priority queue?

    (a) binary search tree
    (b) hash table
    (c) linked list
    (d) queue
    (e) stack

12. Which of the following does the binary heap implement?

    (a) binary search tree
    (b) hash table
    (c) priority queue
    (d) queue
    (e) stack

13. 6, 8, 4, 3, and 1 are inserted into a data structure in that order. An item is deleted using only a basic data structure operation. If the deleted item is a 1, the data structure cannot be a

(a) hash table
(b) priority queue
(c) queue
(d) search tree
(e) stack

14. Which data structure is used by the compiler to implement recursion?

    (a) hash table
    (b) priority queue
    (c) queue
    (d) search tree
    (e) stack

15. Which of the following data structures uses a sentinel?

    (a) binary heap
    (b) hash table
    (c) queue
    (d) stack
    (e) none of the above use sentinels

16. A node with key 8 has a left child with key 10. Which of the following objects could this node be found in?

    (a) binary search tree
    (b) max heap
    (c) min heap
    (d) two of the above
    (e) none of (a), (b), and (c)

17. *Percolate up* and *down* are used for

    (a) AVL trees
    (b) B-trees
    (c) circular queue
    (d) binary heaps
    (e) none of the above

18. Which of the following is true about the skew heap?

    (a) it is balanced
    (b) each node stores nothing besides an item and two pointers
    (c) the right path contains at most a logarithmic number of nodes
    (d) two of the above

(e) all of (a), (b), and (c)

19. Which of the four operations below can be used to implement the other three for the skew heap?

    (a) `Decrease_Key`
    (b) `Delete_Min`
    (c) `Insert`
    (d) `Merge`
    (e) none of the above

20. Which of the following is not a binary tree?

    (a) binary heap
    (b) binomial queue
    (c) skew heap
    (d) splay tree
    (e) all of the above are binary trees

Answers: 1-B, 2-A, 3-A, 4-D, 5-D, 6-B, 7-E, 8-D, 9-B, 10-C, 11-A, 12-C, 13-C, 14-E, 15-A, 16-C, 17-D, 18-B, 19-D, 20-B.

Chapter 7

1. What is the basic algorithm used for external sorting?

    (a) finding the median
    (b) merging
    (c) selection
    (d) all of the above
    (e) none of (a), (b), and (c)

2. Which of the following data structures does not yield an efficient comparison-based sort?

    (a) AVL tree
    (b) hash table
    (c) priority queue
    (d) all can be used for efficient sorting
    (e) none can be used for efficient sorting

3. Which of the following algorithms requires the most extra space, on average, when implemented as in the text?

    (a) heapsort
    (b) insertion sort
    (c) mergesort

(d) quicksort
(e) shellsort

4. Which of the following is the strongest lower bound for sorting when ordering information is obtained only by *adjacent comparisons*?
   (a) $O(n \log n)$
   (b) $O(n^2)$
   (c) $\Omega(n \log n)$
   (d) $\Omega(n^2)$
   (e) none of the above is a valid lower bound for this problem
5. Which of the following algorithms runs in quadratic average time?
   (a) heapsort
   (b) insertion sort
   (c) mergesort
   (d) quicksort
   (e) shellsort
6. Which of the following algorithms runs in $O(n \log n)$ average time but quadratic worst-case time?
   (a) heapsort
   (b) insertion sort
   (c) mergesort
   (d) quicksort
   (e) shellsort
7. Which of the following algorithms, implemented as in the text, runs in $O(n)$ time when presented with an array of n identical elements?
   (a) heapsort
   (b) insertion sort
   (c) mergesort
   (d) quicksort
   (e) shellsort
8. Which of the following algorithms has the largest big-Oh differential between average-case and worst-case performance?
   (a) heapsort
   (b) insertion sort
   (c) mergesort
   (d) quicksort
   (e) quickselect

9. How much extra space is used by heapsort?
   (a) $O(1)$
   (b) $O(\log n)$
   (c) $O(n)$
   (d) $O(n^2)$
   (e) none of the above

10. Which sorting algorithm has the same average and worst-case time bounds (in Big-Oh) as heapsort?
   (a) insertion sort
   (b) mergesort
   (c) quicksort
   (d) shellsort
   (e) none of the above

11. For quicksort, what do I and J do when they see keys equal to the pivot?
   (a) I stops, J stops
   (b) I stops, J goes
   (c) I goes, J stops
   (d) I goes, J goes
   (e) I and J alternate between stopping and going

12. In median-of-three partitioning, where is the pivot placed before partitioning begins?
   (a) at the start of the array
   (b) at the middle of the array
   (c) at the end of the array
   (d) in a temporary variable
   (e) none of the above

13. Which of the following statements about sorting five elements is the strongest statement that is directly implied by the information theoretic lower bound?
   (a) 6 comparisons are sufficient
   (b) 6 comparisons are necessary and sufficient
   (c) 7 comparisons are necessary
   (d) 7 comparisons are sufficient
   (e) 7 comparisons are necessary and sufficient

14. *Replacement selection* is
   (a) arranging the initial runs on the tape in an optimal way
   (b) constructing the runs so they have expected length $2M$

(c) using K-way merging instead of 2-way merging

(d) using $K+1$ tapes instead of K tapes

(e) none of the above

Answers: 1-B, 2-B, 3-C, 4-D, 5-B, 6-D, 7-B, 8-E, 9-A, 10-B, 11-A, 12-E, 13-C, 14-B.

Chapter 8

1. Which of the following trees can have height that is not logarithmic?

   (a) AVL tree

   (b) binary heap

   (c) B-tree of order 4

   (d) union/find tree, with union-by-height

   (e) all of the above trees must have logarithmic depth

2. Which of the following properties is not required for an equivalence relation?

   (a) reflexive

   (b) symmetric

   (c) transitive

   (d) all of these properties are required

   (e) none of these properties is required

3. Which of the following is an equivalence relationship?

   (a) a R b if there is a path from a to b in a directed graph G

   (b) a R b if a and b are two people who know each other

   (c) a R b if a and b end in the same two digits

   (d) all of the above

   (e) none of (a), (b), (c)

4. Which of the following, when performed by itself, is sufficient to ensure a bound of $O(m \log n)$ for m operations?

   (a) path compression

   (b) union by height

   (c) union by size

   (d) all of the above

   (e) none of (a), (b), and (c)

5. *Path compression* is

   (a) performed during `Unions` to make `Unions` faster

   (b) performed during `Unions` to make `Finds` faster

(c) performed during `Finds` to make `Finds` faster

(d) performed during `Finds` to make `Unions` faster

(e) performed during `Finds` to make both `Finds` and `Unions` faster

6. What is the value of $\log^* 65536$?

   (a) 1

   (b) 4

   (c) 16

   (d) 32

   (e) none of the above

Answers: 1-E, 2-D, 3-C, 4-D, 5-C, 6-B.

Chapter 9

1. Which of the following is a synonym for an edge?

   (a) arc

   (b) node

   (c) path

   (d) vertex

   (e) none of the above

2. Which of the following problems is not known to be solvable in linear time?

   (a) topological sort

   (b) unweighted shortest path in general graphs

   (c) weighted shortest path in acyclic graphs

   (d) weighted shortest path in cyclic graphs

   (e) all are solvable in linear time

3. Which of the following does not use a queue?

   (a) negative weighted shortest path algorithm

   (b) positive weighted shortest path algorithm

   (c) topological sort

   (d) unweighted shortest path algorithm

   (e) all of the above use a queue

4. Which of the following algorithms solves the unweighted single source shortest path problem?

   (a) breadth first search

   (b) depth first search

   (c) Dijkstra's algorithm

(d) Kruskal's algorithm
(e) Prim's algorithm

5. Which of the following algorithms solves the positive weighted single source shortest path problem?

   (a) breadth first search
   (b) depth first search
   (c) Dijkstra's algorithm
   (d) Kruskal's algorithm
   (e) Prim's algorithm

6. In a graph with v vertices and e edges, which of the following maximum sizes is not correct for an unweighted shortest path computation?

   (a) v for the number of adjacency lists
   (b) e for the total size of all adjacency lists
   (c) e for the size of the hash table that maps names to internal numbers
   (d) v for the size of the queue
   (e) all of the above are correct

7. In a connected graph with no loops or multiple edges, which of the following inequalities is not correct? (v is the number of vertices, e is the number of edges)

   (a) $e \leq v^2$
   (b) $e \geq v - 1$
   (c) $v \leq e^2 + 1$
   (d) $v \geq e/2$
   (e) all of the above are correct

8. If the shortest path algorithm is run and a vertex is not reachable from the starting point, what happens?

   (a) a distance of infinity is reported
   (b) a distance of -1 is reported
   (c) a distance of zero is reported
   (d) the algorithm enters an infinite loop
   (e) the algorithm's results are undefined

9. For the weighted shortest path problem, let d_v be the cost of reaching the current vertex v, let w be adjacent to v and assume the edge cost is $c_{v,w}$. Suppose that d_w was the cost of reaching w prior to examining v. (Ties are broken in favor of the first path seen). Then under what circumstances is w's distance lowered?

   (a) $d_w > d_v$
   (b) $d_w > d_v + 1$

(c) $d_w > d_v + c_{v,w}$

(d) $d_v > d_w + c_{v,w}$

(e) none of the above

10. Which of the following statements is true?

    (a) A topological ordering exists in every directed graph

    (b) Every acyclic graph has at least one topological ordering

    (c) Every acyclic graph has exactly one topological ordering

    (d) Every acyclic graph has at most one topological ordering

    (e) none of the above

The next four questions refer to the following directed graph: $V = \{V_0, V_1, V_2, V_3, V_4, V_5, V_6\}$. *There are the following twelve edges, with edge costs listed as the third item in the triplet:* $E = \{(V_0, V_2, 4), (V_1, V_0, 2), (V_1, V_3, 3), (V_3, V_0, 1), (V_3, V_2, 2), (V_3, V_5, 8), (V_3, V_6, 4), (V_4, V_1, 10), (V_4, V_3, 2), (V_4, V_6, 7), (V_5, V_2, 2), (V_6, V_5, 1)\}$.

11. The shortest weighted path from V_4 to V_5 has weight

    (a) 2

    (b) 4

    (c) 7

    (d) 8

    (e) none of the above

12. If the start vertex is V_4, then using the standard weighted shortest path algorithm, which is the last vertex to be declared known?

    (a) V_0

    (b) V_1

    (c) V_2

    (d) V_4

    (e) none of the above

13. If the start vertex is V_4, then using the acyclic weighted shortest path algorithm, which is the last vertex to be declared known?

    (a) V_0

    (b) V_1

    (c) V_2

    (d) the graph is not acyclic, so the acyclic algorithm should not be used

    (e) none of the above

14. If the above graph were undirected, then what would be the cost of its minimum spanning tree?

    (a) 1

(b) 10
(c) 11
(d) 12
(e) none of the above

15. Which algorithm is used to compute minimum spanning trees?
    (a) breadth first search
    (b) depth first search
    (c) Dijkstra's
    (d) Kruskal's
    (e) none of the above

Answers: 1-A, 2-D, 3-B, 4-A, 5-C, 6-C, 7-D, 8-A, 9-C, 10-B, 11-C, 12-B, 13-C, 14-B, 15-D.

Chapter 10

1. Which of the following strategies do not directly invoke recursion?
    (a) backtracking
    (b) divide and conquer
    (c) dynamic programming
    (d) two of the above do not directly invoke recursion
    (e) none of (a), (b), and (c) directly invoke recursion
2. 10000 random integers are generated randomly with a uniform distribution over the range 1 to 10000 inclusive. Which of the following would indicate a poor generator?
    (a) the average of the numbers is about 4999
    (b) each number appears exactly once
    (c) no four consecutive numbers are all even
    (d) two of the above
    (e) all of (a), (b), and (c)
3. The *seed* of a linear congruential generator is
    (a) always zero
    (b) occasionally zero, depending on other random events
    (c) the initial value
    (d) the multiplier
    (e) the period of the generator
4. Which of the following is a bad case for randomized quickselect?
    (a) any input with $K = 1$

(b) reverse ordered input

(c) sorted input

(d) there are no bad inputs

(e) none of the above

5. If the randomized primality testing algorithm (with one iteration) declares that P is prime and C composite, then which of the following is most accurate?

   (a) There is at most a 25 percent chance that P has been declared prime falsely and there is at most a 25 percent chance that C has been declared composite falsely

   (b) P is prime with 100 percent certainty but there is at most a 25 percent chance that C has been declared composite falsely

   (c) There is at most a 25 percent chance that P has been declared prime falsely, but C is composite with at least 100 percent certainty

   (d) P is prime with 100 percent certainty and C is composite with 100 percent certainty

   (e) All of the above statements are factually incorrect

Answers: 1-C, 2-D, 3-C, 4-D, 5-C.