Linked Lists

1. Consider a function

```
static void append(LinkedList list1, LinkedList list2)
   // appends list2 to the end of list1
```

What is the best runtime of this function if you assume that both input lists have n elements?

(a) $O(1)$
(b) $O(n)$
(c) $O(n \log n)$
(d) $O(n^2)$

2. Starting with an empty linked list, what is the total cost of adding n elements to the list all at the front of the list?

(a) $O(1)$
(b) $O(n)$
(c) $O(n \log n)$
(d) $O(n^2)$

SEList

3. Consider a space-efficient linked list (SEList) with block size $b = 3$. What is the **maximum** number of non-empty nodes that can be used to store $n = 1,000,003$ elements?

(a) $499,999$
(b) $500,000$
(c) $500,001$
(d) $500,002$
(e) $500,003$

4. Consider a space-efficient linked list (SEList) that has block size $b = 10$ and stores $n = 37$ list elements. Which of the following statements is true about the number of **rugged** nodes in the list?

(a) The number of rugged nodes in the list is at least three.
(b) The number of rugged nodes in the list is at most three.
(c) The number of rugged nodes in the list cannot be three.

5. Consider the algorithm for adding a new list element in node u_0 (that is found from the `getLocation(i)` method) in a space-efficient linked list (SEList) of block size b. If you run off the end of the list of blocks, while looking for a block that is not full, **before** $b - 1$ steps, what is the time complexity associated with this add operation?

(a) $O(\log b)$
(b) $O(b)$

(c) $O(b^2)$
(d) $O(b^3)$

6. Consider a space-efficient linked list (SEList) that has block size b. What is the worst-case time complexity of the `gather` function if the list has n elements?

   (a) $O(b)$
   (b) $O(b \log b)$
   (c) $O(b^2)$

7. Consider a space-efficient linked list (SEList) that has block size b and stores n elements. **Not** including the space needed to store the data (i.e., the n references to the data) and **not** including the space for the dummy node (since it is a doubly-linked list), how much memory is wasted (i.e., not used to store data) in the BEST case?

   (a) There is no wasted space.
   (b) $2n/(b+1)$.
   (c) The remainder of n divided by $b = 1$.

8. Consider a space-efficient linked list (SEList) that has block size $b = 9$ and suppose that every block is rugged except the last block. What is the **local block index** of the element with list index $i = 21$?

   (a) 1
   (b) 2
   (c) 3
   (d) 4
   (e) 5

9. Consider a space-efficient linked list (SEList) with block size $b = 5$. If the blocks are labeled b_k for $k \in \{0, 1, \ldots, r-1\}$, what is the LARGEST value of k that the element with list index $i = 17$ can be?

   (a) 1
   (b) 2
   (c) 3
   (d) 4
   (e) 5

10. Consider a space-efficient linked list (SEList) that has block size b and contains $n = b^2$ list elements. When removing an element from the MIDDLE of the list, in which the `gather()` function is called, which of the following tasks dominates the cost (i.e., accounts for the largest component of the cost of the whole operation)?

    (a) Finding the node where list element i currently resides.
    (b) Shifting elements between nodes to close the resulting gap.
    (c) Shifting element within the bounded dequeue to close the resulting gap.
    (d) Neither of these will dominate the overall cost.

Skip Lists

11. Suppose that promotions of a skiplist were determined by flipping a biased coin for which the probability of heads (i.e., the probability of promoting a node) was 1/3 and the probability of tails was 2/3. What is the expected number of nodes in the skiplist if we ignore occurrences of the sentinel?

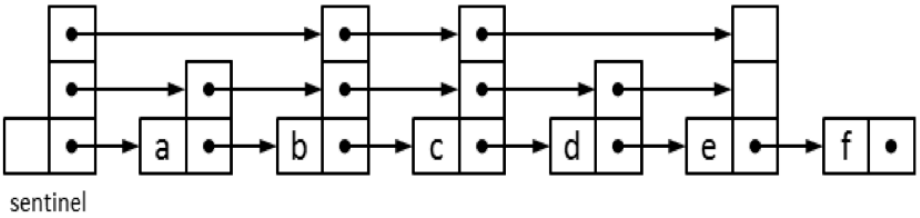
    (a) $2n/3$

    (b) $3n/2$

    (c) $2n$

    (d)

12. In order for a SkiplistList to be an effective implementation of the List interface, the individual nodes of a SkiplistList each contain an array of integers (`int[] length`). Consider the SkiplistList depicted below; if Node c were removed, what would be the contents of the length array of node b?



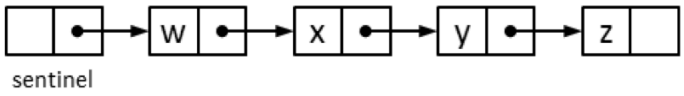
    (a) $[1, 1, 2]$

    (b) $[1, 2, 2]$

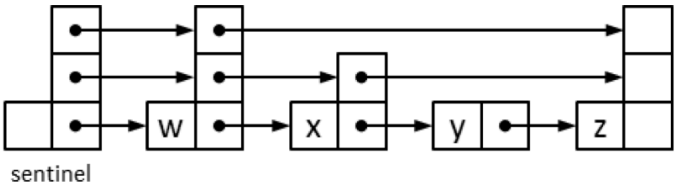
    (c) $[2, 2, 2]$

    (d)

13. When creating a skiplist (using the approach discussed in class and found in the textbook) starting from the following existing singly-linked list



    which of these "coin-flip" sequences would create the following skiplist, if you flip all the coins for L_0, then L_1, etc., and only promote on a "heads"?



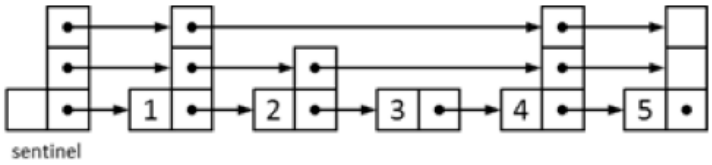
    H denotes heads and T denotes tails.

    (a) H,H,H,T,T,H,H

    (b) H,H,H,T,T,T,H,H

(c) H,H,T,H,H,T,H,T,T

14. Consider the following skiplist:



What sequence of operations would form the search path for the node containing the value 3?

(a) $\rightarrow, \downarrow, \downarrow, \rightarrow, \rightarrow$
(b) $\rightarrow, \downarrow, \rightarrow, \downarrow, \rightarrow$
(c) $\downarrow, \downarrow, \rightarrow, \rightarrow, \rightarrow$

15. Consider a skiplist with n elements that has height $h = \log n$. What is the worst-case time complexity of an operation that adds a new node to a skiplist if the new node has height (number of promotions) 12?

(a) $O(1)$
(b) $O(\log n)$
(c) $O(n)$
(d) $O(n \log n)$
(e) $O(n^2)$

16. Consider a skiplist with n elements that has height $h = n^2$. What is the **expected** worst-case time complexity of an operation that adds a new node to a skiplist if the new node has height (number of promotions) 12 in this particular skiplist?

(a) $O(1)$
(b) $O(\log n)$
(c) $O(n)$
(d) $O(n \log n)$
(e) $O(n^2)$

Hash Tables

17. Consider a hash table with n elements. If the hash table uses chaining with hashing and a good hash function, which operation will run in guaranteed constant time?

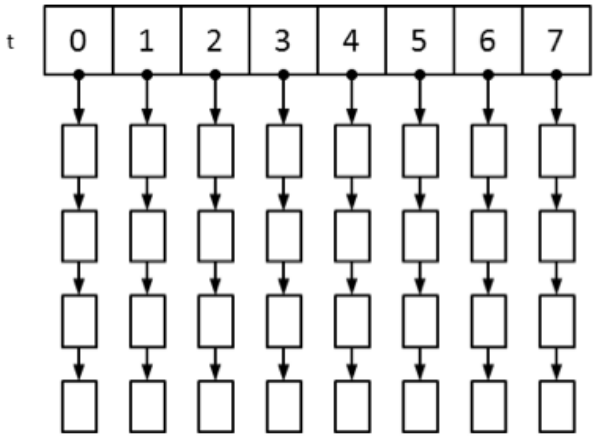
(a) Finding an element.
(b) Adding an element.
(c) Removing an element.
(d) None of these operations can run in constant time.

18. Consider a hash table with n elements and bucket array of capacity M. If the hash table uses chaining with hashing and a good hash function, what is the **expected** size of each bucket when $n = O(M^2)$?

(a) $O(1)$
(b) $O(\log n)$
(c) $O(\log M)$
(d) $O(M)$

19. Consider a ChainedHashTable (a hash table using chained hashing from the textbook) with dimension $d = 3$ and hash function $h(x) = (x - 1) \mod 8$. Suppose that keys $19, 30, 11, 16, 24$, and 27 have been added to the hash table. What is the size of the largest bucket in the hash table?

    You can use the following diagram to help with your solution.



(a) 1
(b) 2
(c) 3
(d) 4

20. Consider a hash table with n elements. What is the worst-case runtime of finding an element if the hash table uses linear probing?

(a) $O(1)$
(b) $O(\log n)$
(c) $O(n)$
(d) $O(n \log n)$

21. Consider a LinearHashTable (a hash table using linear probing from the textbook) with dimension $d = 3$ using the hash function $h(x) = (x \mod 4) \times 2$.

    What is longest consecutive sequence of **NUL** values (including wrapping around the end of the bucket array) after the following sequence of operations is performed (starting with an empty hash table)?

```
add( 13 )
add( 9 )
add( 23 )
add( 7 )
add( 15 )
```

You can use the following table to help with your work. (Yes, this is a bad hash function.)

0	1	2	3	4	5	6	7

(a) 1

(b) 2

(c) 3

(d) 4

22. Consider a cuckoo hash table where each of the arrays has capacity 8 and the hash functions are given by

$$h_1(x) = (x \bmod 4) \times 2$$

$$h_2(x) = (3x) \bmod 8$$

Which of the following accurately represents the state of the hash table after adding the elements 3, 10, 7, and 14 (in that order)?

(a) table 1 :

	0	1	2	3	4	5	6	7
Table 1:					19		3	
Table 2:			14			7		

(b) table 1 :

	0	1	2	3	4	5	6	7
Table 1:					14		7	
Table 2:		3					10	

(c) table 1 :

	0	1	2	3	4	5	6	7
Table 1:			14		10	7	3	
Table 2:								

Binary Trees

Consider the following binary tree.

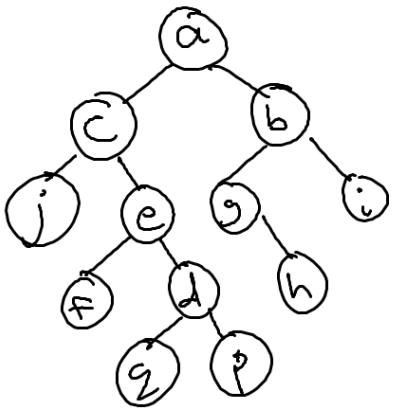


Figure 1: Binary Tree

23. What is the height of the tree shown in Figure 1?

    (a) 2
    (b) 3
    (c) 4
    (d) 5

24. What is the height of vertex **d** in the tree shown in Figure 1?

    (a) 2
    (b) 3
    (c) 4
    (d) 5

25. What is the depth of vertex **d** in the tree shown in Figure 1?

    (a) 2
    (b) 3
    (c) 4
    (d) 5

26. What is the minimum height of a binary tree with $n = 13$ elements?

    (a) 2
    (b) 3
    (c) 4

27. Consider a preorder traversal of the tree in Figure 1 that simply displays the contents of each node. Which of the following is the correct sequence that is output?

    (a)
    (b)
    (c)

Binary Search Trees

28. The BinarySearchTree (from the textbook) implements the SSet interface. What is the **expected** worst-case runtime of the `add(x)`, `remove(x)`, and `find(x)` operations in a BinarySearchTree if the tree contains n nodes?

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

29. Consider the binary search tree created with the following permutation:

    $$< 2, 5, 9, 3, 7, 6, 4, 8, 1, 0, 10 >$$

    Which of the following nodes is on the search path for 8?

(a) The node containing 3
(b) The node containing 6
(c) The node containing 7

30. Consider the binary search tree created with the following permutation:

$$< 2, 5, 9, 3, 7, 6, 4, 8, 1, 0, 10 >$$

What is the height of resulting binary search tree?

(a) 2
(b) 3
(c) 4
(d) 5

31. Consider the binary search tree created with the following permutation:

$$< 2, 5, 9, 3, 7, 6, 4, 8, 1, 0, 10 >$$

What is the depth of the node with 7 in the resulting binary search tree?

(a) 2
(b) 3
(c) 4
(d) 5

32. In a binary search tree created from a random permutation of the first 20 non-negative integers (i.e., values from the range [0, 19]), what is the probability that the node associated with a value of 3 would be found on the search path for the node associated with a value of 18?

(a) $\frac{1}{14}$
(b) $\frac{1}{15}$
(c) $\frac{1}{16}$
(d) $\frac{1}{17}$
(e) $\frac{1}{18}$

33. In a binary search tree created from a random permutation of the first 20 non-negative integers (i.e., values from the range [0, 19]), what is the probability that the node associated with a value of 13 would be found on the search path for the node associated with a value of 7.5?

Please note that 7.5 does NOT currently appear anywhere in the tree.

(a) 1/4
(b) 1/5
(c) 1/6
(d) 1/7
(e) 1/8