Indiana University Bloomington

Fall-2023
CSCI-B505 / INFO-I500
APPLIED ALGORITHMS
Examination – I

September 19, 2023, Tuesday, 6:45 PM – 7:45 PM.

Name & Surname	
University ID	
Signature	

Rules:

1. There are 25 questions in this examination.
2. Duration of the exam is 60 minutes.
3. Write your name and surname on every page at the designated positions.
4. Put your ID card on your desk so that the proctors can check your identity.
5. The use of lecture notes, books, and any other resources, calculators, computers, mobile phones, and any digital equipment is prohibited.
6. Every student taking this examination is subject to the university discipline code. Any act or attempt of cheating, including helping others, will be considered a violation of the code.

Name & Surname:

1. Given the function $f(x) = log(x) + 500$, which function for $g(x)$ would make $f(x) \in \omega(g(x))$.

   (a) $g(x) = x$
   (b) $g(x) = log(log(x))$ ✓
   (c) $g(x) = log(x)$
   (d) $g(x) = x/2$

2. Which of the following relationships are correct:

   I) $f(n) \in O(g(n)) \iff g(n) \in \Omega(f(n))$
   II) $f(n) \in \Theta(g(n)) \iff g(n) \in \Theta(f(n))$
   III) $f(n) \in \Theta(g(n)) \iff f(n) \in O(g(n))$ AND $f(n) \in \Omega(g(n))$
   IV) $f(n) \in O(f(n))$

   (a) Only I
   (b) I and III
   (c) I, II, and III
   (d) All the above ✓

3. Which of the following is not $O(n^2)$?

   (a) $15 * (n^2)$
   (b) $n^{1.98}$
   (c) $(n^3)/(sqrt(n))$ ✓
   (d) $2.5 * (n^2)$

4. For the functions $log_2 n$ and $log_8 n$, what is the asymptotic relationship between these functions?

   (a) $log_2 n \in O(log_8 n)$
   (b) $log_2 n \in \Omega(log_8 n)$
   (c) $log_2 n \in \Theta(log_8 n)$
   (d) All of the above ✓

5. While calculating the time complexity of an algorithm through experimental analysis we have observed the following readings for different inputs of n.

   I) For $n = 10$, the time taken is 5.05 ms

   II) For $n = 20$, the time taken is 15.5 ms

   III) For $n = 40$, the time taken is 44.6 ms.

   IV) For $n = 80$, the time taken is 127.3 ms.

   Based on these readings, what do you think best describes the time complexity of the algorithm?

   (a) $O(logn)$

   (b) $O(n^2)$

   (c) $O(nlogn)$ ✓

   (d) $O(n^3)$

6. What is the worst-case time complexity for the below mystery function:

```
def mystery_function(n):
    i = 1
    total_sum = 0
    while i < n:
        j = 1
        while j < n:
            k = 1
            while k < n:
                total_sum += (i + j + k)
                k *= 2
            j *= 3
        i *= 4
```

   (a) $O(n^3)$

   (b) $O(log(n) * log(n) * log(n))$

   (c) $O(log_4(n) * log_3(n) * log_2(n))$ ✓

   (d) $O(log_4(n) + log_3(n) + log_2(n))$

7. Tom is a computer science student working on a programming project that involves processing a large dataset. While working on his project, Tom notices that his program frequently accesses the same data elements multiple times within a short time frame. Which type of locality of reference is observed?

   (a) Temporal Locality ✓

   (b) Spatial Locality

   (c) Both A and B

   (d) None of the Above

8. Given a skip list of 32 elements. What is the expected number of nodes at level 3? Assume the skip list starts from level 0 and each element in the list is carried to the next level with a probability of 0.5.

   (a) 2

   (b) 4 ✓

   (c) 8

   (d) 16

9. A Range Minimum Query (RMQ) is a fundamental algorithm used to find the minimum element within a range of values in an array. We have studied that the most efficient implementation of this algorithm runs in $O(1)$ time and $O(nlog(n))$ space complexity. Suppose we have an unknown array A with 9 elements **where index starts from 1** and we are given the matrix M where each cell $M[i, j]$ of the matrix stores the value $min(A[i...(i+2^j-1)])$.The RMQ(l, r) returns $min(A[l.....r])$.What shall be the value of RMQ(3,7)?

$M[i, j]$	1	2	3
1	8	8	8
2	8	8	5
3	23	23	X
4	23	23	X
5	31	17	X
6	36	5	X
7	17	X	X
8	5	X	X

Note: 'X' in the matrix M means value not defined.

   (a) 22

   (b) 13

   (c) 17

   (d) 23 ✓

10. Insertion of an element at the middle of a singly linked list with the number of nodes n $(n > 11)$ requires the modification (includes updates and new links) of how many pointers?

    (a) 1

    (b) $n/2$

    (c) 2 ✓

    (d) n

11. Which of the following points is/are true about Linked List data structure when it is compared with the array?

    I) Arrays have better space locality that can make them better in terms of performance.

    II) It is easier to insert and delete elements in an array than in a linked list.

    III) Random access is not allowed in a typical implementation of Linked Lists

    IV) The size of the array has to be pre-decided, linked lists can change their size at any time.

    (a) I, II, III are true

    (b) I, III, IV are true ✓

    (c) Only I and IV are true

    (d) All are true

12. Consider an unsorted singly linked list, which has both head and tail pointers. Then which of the following can be done in $O(1)$ time complexity?

    I) Inserting an element at the beginning of the linked list

    II) Deleting the first element of the linked list

    III) Inserting an element at the end of the linked list

    IV) Deleting the last element of the linked list

    Choose the correct option:

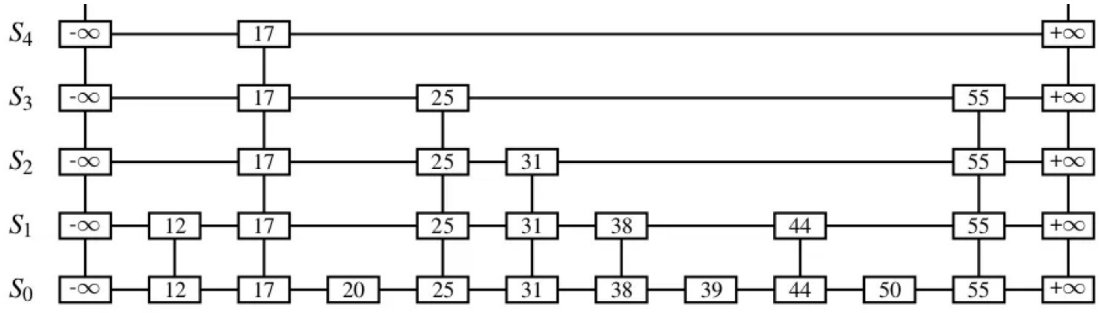
    (a) I and II are correct

    (b) II and IV are correct

    (c) I, II, and III are correct ✓

    (d) All are correct

13. In the given Skip list what is the order of the elements that are accessed while searching for element 50?



    (a) $-\infty, 17, 25, 31, 38, 39, 44, 50$

    (b) $-\infty, 17, +\infty, 25, 55, 31, 55, 38, 44, 55, 50$ ✓

    (c) $-\infty, 17, +\infty, 20, 25, 31, 38, 39, 44, 50$

    (d) $-\infty, 12, 17, 20, 25, 31, 38, 39, 44, 50$

14. For the given implementation of queue, if we do the following operations in order. How many times stack push and pop functions will be called respectively?

```
enQueue(q,23)
enQueue(q,7)
deQueue(q)
enQueue(q,2)
enQueue(q,16)
```

```
1     def enQueue(q, x):
2         while(!st1.empty()):
3             st2.push(st1.top())
4             st1.pop()
5         st1.push(x)
6
7         while(!st2.empty()):
8             st1.push(st2.top())
9             st2.pop()
10
11    def deQueue(q):
12        if(st1.empty()):
13            print("Queue is empty")
14        else:
15            print("The first in queue is" + st1.top())
16            st1.pop()
```

(a) 10, 8

(b) 10, 9

(c) 12, 10

(d) 12, 9 ✓

15. Which of the following statements are correct?

(a) Stack is a non-linear data structure

(b) Stack is used to solve breadth-first traversal in a tree

(c) Queue is a FIFO (First In First Out) data structure ✓

(d) All the above

16. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (()(())(())))?

    (a) 1

    (b) 2

    (c) 3 ✓

    (d) 4 or more

17. A stack, of max size n, is created using an array A[0...n-1] and a variable top. The push and pop for the stack are implemented as follows.

```
push(x):
    A[top] <- x
    top <- top - 1

pop:
    top <- top + 1
    return A[top]
```

    What will be the value of the top when the stack is empty and full respectively? And also how will you calculate the size of the stack?

    (a) top = -1, top = n, size = top

    (b) top = -1, top = n-1, size = top+1

    (c) top = n-1, top = -1, size = (n-top)-1 ✓

    (d) top = n-1, top = -1, size = (n-top)

18. Suppose an initially empty queue Q has executed a total of 22 enqueue operations, 6 first operations, and 10 dequeue operations, 4 of which raised empty errors that were caught and ignored. What is the current size of Q?

    - enqueue: Adds an element to the queue
    - first: Returns an element from the queue without removing it
    - dequeue: Removes and returns an element from the queue and throws an error when the queue is empty

    (a) 14

    (b) 15

    (c) 16 ✓

    (d) 17

19. Which of the following statements is true about the linked list implementation of the Queue?

    (a) In a push operation, if new nodes are inserted at the beginning of the linked list, then in a pop operation, nodes must be removed from the end.

    (b) In a push operation, if new nodes are inserted at the end, then in a pop operation, nodes must be removed from the beginning.

    (c) Both (a) and (b) ✓

    (d) None of the above

20. After inserting the elements 55, 67, 26, 37, 61, and 35 into an empty binary search tree in the sequence shown, the element in the lowest level is

    (a) 35 ✓

    (b) 26

    (c) 55

    (d) 37

21. For this postfix expression, $abcd-*+e-$, what will be the prefix expression?

    (a) $-+b*a-cde$

    (b) $-+a*b-dce$

    (c) $-+a*b-cde$ ✓

    (d) None of the above

22. If the preorder traversal of a binary tree is $[B,A,D,C,E]$ and the postorder traversal is $[A,C,E,D,B]$, what is the corresponding inorder traversal?

    (a) $[A,B,C,E,D]$

    (b) $[B,A,C,D,E]$

    (c) $[A,B,C,D,E]$ ✓

    (d) $[B,A,C,E,D]$

23. The following three are known to be the preorder, inorder, and postorder sequences of a binary tree. But it is not known which is which.

    I MBCAFHPYK

    II KAMCBYPFH

    III MABCKYFPH

    Choose the correct option:

    (a) I and II are preorder and inorder sequences, respectively

    (b) I and III are preorder and postorder sequences, respectively

    (c) II is the inorder sequence, but nothing more can be said about the other two

    (d) II and III are the preorder and inorder sequences, respectively ✓

24. Given a full binary tree where every internal node in the tree has exactly two children. This tree with 2n+1 total number of nodes contains:

    (a) $n-1$ leaf nodes

    (b) n leaf nodes

    (c) $n-1$ internal nodes

    (d) n internal nodes ✓

25. The pre-order traversal of a binary search tree is `[14, 8, 7, 10, 15, 17]`. Which of the following can be the post-order traversal of the same binary search tree?

    (a) `[7, 10, 8, 17, 15, 14]` ✓

    (b) `[7, 10, 14, 8, 17, 15]`

    (c) `[7, 10, 8, 15, 17, 14]`

    (d) `[7, 8, 10, 17, 15, 14]`