

CS2105 Practice Set — Midterm Preparation

Practice only. These are not the examination questions.

Data Structure and Algorithms with Java (CS2105) · 40 multiple-choice questions · Suggested time: 100 minutes

How to use this set

1. Answer every question without looking at the answers, then check them with the answer sheet.
2. Choose the single best answer (A, B, C or D).

Introduction

1. Which of these is a non-linear data structure?

- A. Array
- B. Tree
- C. Stack
- D. Queue

2. What does this Java statement print?

```
System.out.println(11 % 4);
```

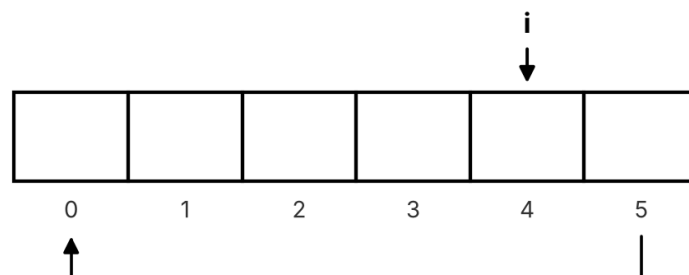
- A. 2.75
- B. 3
- C. 2
- D. 0

3. What does this Java statement print?

```
System.out.println(7.0 / 2);
```

- A. 3.0
- B. 4
- C. 3.5
- D. 3

4. The circular array below has 6 slots. Index i starts at the marked slot. What is i after $i = (i + 3) \% 6$; runs once?



after the last index comes index 0

- A. 7
- B. 0
- C. 1
- D. 3

5. A program often inserts and deletes items in the middle of a list. The number of items is not known in advance, and reading item number i is rare. Which structure is the better choice?
- A. An array, because its size is fixed
 - B. A linked list, because it changes links instead of shifting elements
 - C. A linked list, because it gives $O(1)$ access by index
 - D. An array, because it gives $O(1)$ access by index

Algorithm analysis

6. Which list orders these growth rates from slowest-growing to fastest-growing: $O(n)$, $O(1)$, $O(n \log n)$, $O(\log n)$?
- A. $O(1) < O(\log n) < O(n \log n) < O(n)$
 - B. $O(\log n) < O(1) < O(n) < O(n \log n)$
 - C. $O(1) < O(n) < O(\log n) < O(n \log n)$
 - D. $O(1) < O(\log n) < O(n) < O(n \log n)$
7. An algorithm performs $T(n) = 5n + 100$ operations. What is its tightest Big-O bound?
- A. $O(n^2)$
 - B. $O(1)$
 - C. $O(100)$
 - D. $O(n)$

8. What is the tightest Big-O time bound of this loop?

```
for (int i = 0; i < n; i += 2)
    sum++;
```

- A. $O(n)$
- B. $O(n^2)$
- C. $O(\log n)$
- D. $O(1)$

9. What is the tightest Big-O time bound of this loop?

```
for (int i = 1; i < n; i *= 2)
    sum++;
```

- A. $O(n)$
- B. $O(1)$
- C. $O(n^2)$
- D. $O(\log n)$

10. How many times is `count++` executed in total?

```
int count = 0;
for (int i = 0; i < 5; i++)
    count++;
for (int j = 0; j < 4; j++)
    count++;
```

- A. 9
- B. 20
- C. 5
- D. 4

i	j	count	count++
0	0	1	✓
1	1	2	✓
2	2	3	✓
3	3	4	✓
4	4	5	✓

8

11. How much extra memory (auxiliary space) does this code use, as a function of n ? Do not count the input array a .

```
Algo(a,n){
  int[] b = new int[2 * n];
  for (int i = 0; i < n; i++) {
    b[i] = a[i];
    b[n + i] = a[i];
  }
}
```



$2n + 1$

- A. $O(n^2)$
- B. $O(1)$
- C. $O(2)$
- D. $O(n)$**

12. Which operation on an array of n elements takes $O(1)$ time?

- A. Comparing every pair of elements
- B. Printing every element
- C. Finding the largest element
- D. Reading $a[0]$**

13. What is the tightest Big-O time bound of this fragment?

```
for (int i = 0; i < n; i++) □ n
  for (int j = 0; j < 10; j++) □ 10
    sum++;
```

$T(n) = 10 * n = 10n$

- A. $O(10n^2)$
- B. $O(n^2)$
- C. $O(n)$**
- D. $O(1)$

Recursion

14. What does $f(4)$ return?

```
static int f(int n) {
  if (n == 0) return 0;
  return 2 + f(n - 1);
}
```

A. 8

- B.
- C.
- D. 10

$F(4) = 2 + f(3) == 2 + 6 = 8$
 $F(3) = 2 + f(2) = 2 + 4 = 6$
 $F(2) = 2 + f(1) == 2 + 2 = 4$
 $F(1) = 2 + f(0) == 2 + 0 = 2$
 $F(0) = 0$

15. What is printed by $p(3)$?

```
stat
P(3) 3
P(2) 2
P(1) 1
P(0)
}
```

- A. 3

- B. 3 2 1 0
- C. 1 2 3
- D. 3 2 1**

15. What is printed by p(3)?

```
static void p(int n) {  
    if (n == 0) return;  
    p(n - 1);  
    System.out.print(n + " ");  
}
```

P(3)
P(2)
P(1)
P(0)
1 2 3

A. 3

B. 3 2 1 0

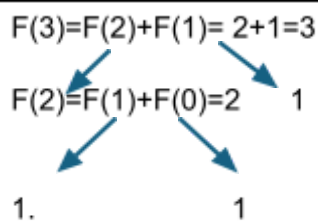
C. 1 2 3

D. 3 2 1

16. How many calls of fib happen in total for fib(3), counting the first call?

```
static int fib(int n) {  
    if (n < 2) return n;  
    return fib(n - 1) + fib(n - 2);  
}
```

A. 3
B. 5



C. 8

D. 4

17. What happens when countDown(3) is called?

```
static void countDown(int n) {  
    System.out.println(n);  
    countDown(n - 1);  
}
```

A. It does not compile

B. It prints 3 2 1 and stops

C. It prints 3, 2, 1, 0, -1, ... and finally throws a StackOverflowError

D. It prints 3 only

18. The method should return n! ($n \times (n - 1) \times \dots \times 1$). What does product(4) actually return?

```
static int product(int n) {  
    if (n == 1) return 1;  
    product(n - 1);  
    return n;  
}
```

A. 4

B. 10

C. 1

D. 24

19. A recursive method makes at most one recursive call each time it runs. Which type of recursion is this?

- A. Indirect recursion
- B. Multiple recursion
- C. Binary recursion
- D. Linear recursion

20. This method should return the sum of $a[i]$, $a[i + 1]$, ..., up to the last element.

Which line belongs at the marked position?

```
static int sumFrom(int[] a, int i) {
    // base case goes here
    return a[i] + sumFrom(a, i + 1);
}
```

- A. if (i == a.length) return a[i];
- B. if (i == a.length) return 0;
- C. if (i == 0) return 0;
- D. if (i == a.length - 1) return 0;

Analysing recursive algorithms and searching

21. A recursive method satisfies $T(n) = T(n-1) + 4$, with $T(0) = 1$. What is $T(2)$?

- A. 8
- B. 5
- C. 9
- D. 12

$$T(2) = T(1) + 4 = 5 + 4 = 9$$

$$T(1) = T(0) + 4 = 1 + 4 = 5$$

22. An algorithm halves the problem size and does a constant amount of work at each step: $T(n) = T(n/2) + c$. What is its running time?

- A. $O(\log n)$
- B. $O(n)$
- C. $O(1)$
- D. $O(n^2)$

23. Binary search looks for 40 in the sorted array below, comparing the middle element of the part still being searched.

Which elements are compared, in order?

3	9	14	22	31	40	55
0	1	2	3	4	5	6

- A. 22, 40
- B. 40
- C. 22, 31, 40
- D. 3, 9, 14, 22, 31, 40

$$\frac{0+6}{2} = 3$$

$$\frac{4+6}{2} = 5$$

24. Binary search is used on a sorted array of about 1,000,000 elements. About how many comparisons does it need in the worst case?

- A. About 1,000
- B. About 20
- C. About 500,000
- D. About 1,000,000

25. An algorithm performs n operations, and each takes 10^{-6} seconds. About how long does it take when $n = 2,000,000$?

- A. 0.2 seconds
- B. 20 seconds
- C. 2,000,000 seconds
- D. 2 seconds

26. Linear search looks for a value that is not in an unsorted array of 50 elements. How many elements does it compare?

- A. 1
- B. 6
- C. 50
- D. 25

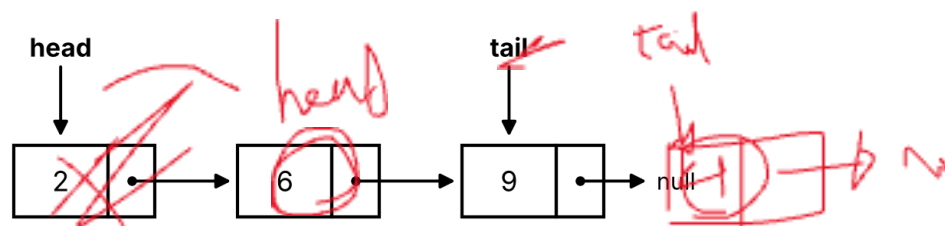
27. A program stores every answer it has already computed in a table,

so a repeated question is answered at once. What does this choice trade?

- A. It uses more memory to save time
- B. It uses less memory and less time
- C. It uses more memory and more time
- D. It saves memory by spending more time

Linked lists

28. L is the lecture's `SinglyLinkedList<Integer>`, shown below. After these calls, what do `L.first()` and `L.last()` return?



```
L.removeFirst();  
L.addLast(4);
```

- A. ~~first() = 2; last() = 9~~ first() = 6; last() = 4
- B. first() = 2; last() = 4
- C. first() = 6; last() = 9
- D. first() = 4; last() = 9

29. Why is `removeLast()` $O(n)$ in a singly linked list that has head and tail references?

- A. Every node must be shifted left
- B. It must walk from head to find the node before tail
- C. tail cannot be read in $O(1)$
- D. The size must be counted again

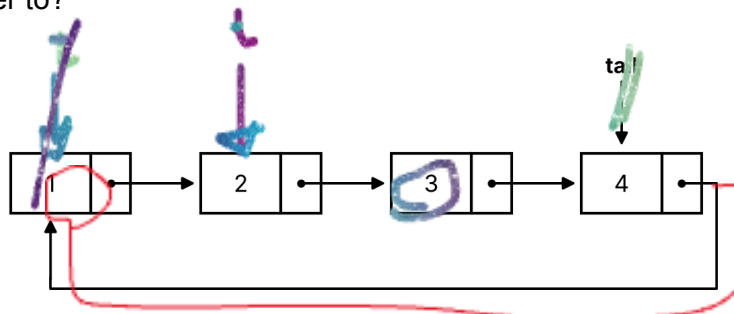
30. In the lecture's `removeFirst()` for a singly linked list, which line belongs in the blank?

```
public E removeFirst() {
    if (isEmpty()) return null;
    E answer = head.getElement();
    head = head.getNext();
    size--;
    if (size == 0)
        _____
    return answer;
}
```



- A. `tail = head.getNext();`
- B. `head = null;`
- C. `tail = null;`
- D. `size = 0;`

31. In the lecture's `CircularlyLinkedList` shown below (only tail is stored), what does `tail.getNext()` refer to?



- A. The node holding 4, the tail itself
- B. The node holding 3
- C. The node holding 1, the first element
- D. null

32. C is the lecture's `CircularlyLinkedList<Integer>` shown in the previous question (1, 2, 3, 4). After `C.rotate(); C.rotate();`, what does `C.first()` return?

- A. 2
- B. 4
- C. 1
- D. 3

33. D is an empty lecture `DoublyLinkedList<Integer>` with header and trailer sentinels. After `D.addLast(7);`, what is `header.getNext()`?

- A. The node holding 7
- B. The header itself
- C. The trailer
- D. null

34. A music app must move to the next song and back to the previous song by following stored links. Which list fits best?

- A. Circularly linked list with only next links
- B. Singly linked list without a tail reference
- C. Doubly linked list
- D. Singly linked list

Stacks, queues and dequeues

35. S is an empty stack. After these calls, what does S.top() return?

```
S.push(1);  
S.push(2);  
S.pop();  
S.push(3);
```

- A. 2
- B. 3
- C. 1
- D. null

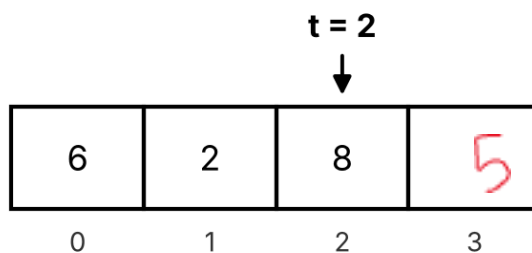
36. Q is an empty queue. After these calls, what does Q.first() return?

```
Q.enqueue("A");  
Q.enqueue("B");  
Q.dequeue();  
Q.enqueue("C");
```

- A. null
- B. B
- C. A
- D. C

37. S is the lecture's ArrayStack<Integer> with capacity 4, shown below. What happens when these calls run?

data (capacity 4)



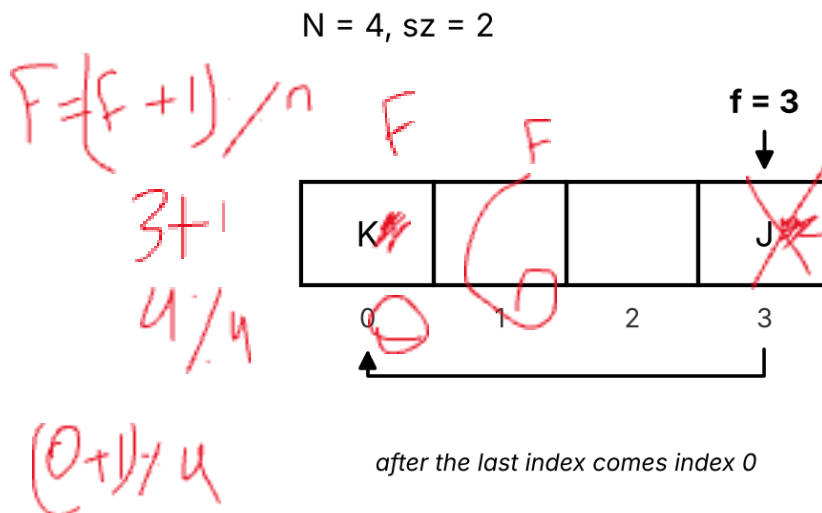
```
S.push(5);  
S.push(1);
```

- A. push(1) overwrites the 6 at index 0
- B. Both pushes work
- C. push(5) works; push(1) throws IllegalStateException
- D. push(5) throws IllegalStateException

38. Using the lectures' stack-based balancing-symbols algorithm, which string is NOT balanced?

- A. `[{()}]`
- B. `{[(())]}`
- C. `()[]{}`
- D. `(([]))`

39. Q is the lecture's `ArrayQueue<String>` with `N = 4`, shown below (`f = 3`, `sz = 2`). After `Q.dequeue()`; `Q.dequeue()`; , what are `f` and `sz`?



- A. `f = 1; sz = 0`
- B. `f = 5; sz = 0`
- C. `f = 0; sz = 0`
- D. `f = 1; sz = 2`

40. D is an empty deque. After these calls, what does D hold, from front to rear?

```
D.addFirst(4);
D.addLast(6);
D.addFirst(2);
D.removeLast();
```

- A. `[6, 4, 2]`
- B. `[2, 4, 6]`
- C. `[4, 6]`
- D. `[2, 4]`

End of practice set.