Indian Institute of Technology Madras, Zanzibar Campus

Z5007 – Programming and Data Structures

QUIZ 1 PRACTICE SET (40 questions, 90 marks) | Data Science & AI | Weeks 1–6

Same format as Quiz 1: Section A (Q1–Q15, multiple choice), Section B (Q16–Q30, code and tracing), Section C (Q31–Q40, analysis and design). Closed book: no notes or calculators. Suggested time: 90 minutes.

Section A: Multiple Choice [21 marks] *One correct answer each.*

Q1. A queue can be built on a circular array or on a linked list, and the user of enqueue and dequeue cannot tell which. This shows that a queue is: **[1]**

(A) an abstract data type with several implementations (B) an algorithm (C) a recurrence (D) only an array

Q2. A procedure loops forever on some valid inputs. Which property of an algorithm does it break? **[1]**

(A) definiteness (B) finiteness (C) input (D) output

Q3. Which function is **not** $O(n^2)$? **[1]**

(A) $7n + 3$ (B) $n \log n$ (C) $n^2/2$ (D) n^3

Q4. For $f(n) = 2n + n \log_2 n$, which statement is true? **[1]**

(A) $f(n) = \Theta(n)$ (B) $f(n) = \Theta(n \log n)$ (C) $f(n) = \Theta(n^2)$ (D) $f(n) = O(\log n)$

Q5. For $n = 1024$, which of these takes the most steps? **[1]**

(A) $100n$ (B) $n \log_2 n$ (C) $n^2/2$ (D) $5000 \log_2 n$

Q6. Binary search is run on a sorted list of 1000 items. At most how many items are compared with the key? **[2]**

(A) 5 (B) 10 (C) 500 (D) 1000

Q7. Items are appended one by one to a dynamic array that doubles when full. The average cost per append over n appends is: **[2]**

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

Q8. Insertion sort is run on a list that is already sorted, n items. How many comparisons does it make? **[1]**

(A) $n - 1$ (B) $n(n-1)/2$ (C) $n \log n$ (D) 0

Q9. Which postfix expression equals the infix expression $(3 + 4) \times 5$? **[2]**

(A) `3 4 5 + ×` (B) `3 4 + 5 ×` (C) `3 + 4 5 ×` (D) `× + 3 4 5`

Q10. A stack starts empty. Operations: push 1, push 2, push 3, pop, push 4, pop, pop. In what order are the values popped? **[2]**

(A) 3, 4, 2 (B) 3, 2, 4 (C) 1, 2, 3 (D) 3, 4, 1

Q11. A recursive algorithm obeys $T(n) = 2T(n/2) + 1$. Which bound is the tightest? **[2]**

(A) $O(1)$ (B) $O(\log n)$ (C) $O(n)$ (D) $O(n \log n)$

Q12. Print jobs are served in the order in which they arrive. The best structure is a: **[1]**

(A) stack (B) queue (C) sorted array (D) heap

Q13. A circular-array queue has `cap = 8` and `head = 5`. Four dequeues are done. What is `head` now? **[1]**

(A) 9 (B) 1 (C) 0 (D) 4

Q14. A singly linked list has only a `head` pointer. What is the cost of removing the LAST node? **[1]**

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

Q15. Quicksort picks a uniformly random pivot. For any input, its expected running time is: **[2]**

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

Section B: Code and Tracing [39 marks]

Q16. Bubble sort. Bubble sort (swap if `a[j] > a[j+1]`, each pass shortened by one) runs on `a = [9, 4, 7, 3]`. Write the list after pass 1, the total number of swaps, and the link between that number and the inversions of the input. **[3]**

Q17. Selection sort. Selection sort runs on `a = [64, 25, 12, 22, 11]`. Write the list after pass 1 and after pass 2, and say why selection sort is not stable. **[3]**

Q18. Insertion sort code. Fill in the blanks. **[3]**

```
def insertion_sort(a):
    for i in range(1, len(a)):
        key = a[i]
        j = i - 1
        while ______(1)______:
            ______(2)______
            j -= 1
        ______(3)______
```

Q19. Binary search trace. Binary search (`lo <= hi`, `mid = (lo + hi)//2`) looks for 23 in `a = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]`. Write `lo`, `hi`, `mid` and `a[mid]` for each step. **[3]**

Q20. Linear v binary search. A sorted list has 1,000,000 items. About how many items are compared in the worst case by (a) linear search and (b) binary search? Why does binary search need sorted data? **[3]**

Q21. Stability in merge. In merge sort the merge step takes the left item when `left[i] <= right[j]`. Merge `[(2,a)]` and `[(2,b)]` (sorting by the number only) with `<=` and then with `<`. What changes? **[3]**

Q22. Merge cost. Two sorted lists of sizes 5 and 3 are merged. Give the largest and the smallest possible number of comparisons, with an example of each. **[2]**

Q23. Lomuto partition. Lomuto partition (pivot = last item) is applied to `a = [8, 3, 6, 2, 7, 5]`. Write the array afterwards, the pivot index, and how many swaps were made (including the last one). **[3]**

Q24. Quickselect. Using Q23, find the 3rd smallest item (index 2 in sorted order). Which part is searched next? What about the 5th smallest? **[2]**

Q25. Brackets with a stack. A stack checks the brackets of `([])`: push every opening bracket, and for a closing bracket pop and compare. Show the stack after each symbol and the result. **[2]**

Q26. Circular queue bug. A programmer writes `tail = tail + 1` (no `% cap`) in `enqueue`, with `cap = 4`. The operations enqueue, enqueue, enqueue, enqueue, dequeue, dequeue, enqueue are done. What goes wrong, and what is the fix? **[3]**

Q27. Reversing a linked list. Fill in the blanks, then give the result for $1 \to 2 \to 3 \to 4$. **[3]**

```
prev, curr = None, head
while curr is not None:
    nxt = curr.next
    ______(1)______
    ______(2)______
    ______(3)______
head = prev
```

Q28. Linked-list queue: dequeue. Fill in the blanks (the queue keeps `head` and `tail`, and is not empty). **[3]**

```
def dequeue(self):
    x = ______(1)______
    self.head = ______(2)______
    if self.head is None:
        ______(3)______
    return x
```

Q29. Recursion trace. What does `h(13)` print, and what does the function compute? **[2]**

```
def h(n):
    if n == 0:
        return
    h(n // 2)
    print(n % 2, end="")
```

Q30. Tower of Hanoi. For 5 discs, which disc moves on move number 20? **[2]**

Section C: Analysis and Design [30 marks]

Q31. Big-O proof. Using the definition, prove $2n^3 + 5n^2 + 1 = O(n^3)$. State c and n_0. **[3]**

Q32. Loop analysis. Tightest Big-O of the code, and the reason. **[3]**

```
i = 1
while i < n:
    for j in range(n): pass
    i = i * 3
```

Q33. Loop analysis. Tightest Big-O, and which part dominates. **[3]**

```
for i in range(n):
    for j in range(10): pass
for i in range(n):
    for j in range(n): pass
```

Q34. Choosing a sort. Choose from insertion sort, merge sort and quicksort (random pivot). (a) 12 items on a tiny sensor with almost no memory. (b) A sorted list of 100,000 items that receives one new entry. (c) 10 million items where the worst case must never exceed $O(n \log n)$ and extra memory is available. **[3]**

Q35. Master Theorem. Solve $T(n) = 8\,T(n/2) + n^3$. Give a, b, d, the case and the Big-O. **[3]**

Q36. Writing recurrences. Write and solve the recurrence for (a) binary search and (b) merge sort. **[3]**

Q37. Unrolling. Solve $T(n) = T(n-1) + 3$, $T(0) = 2$ by unrolling, and give the Big-O. **[3]**

Q38. Hanoi proof. The moves of Tower of Hanoi obey $M(n) = 2M(n-1) + 1$, $M(1) = 1$. Prove by induction that $M(n) = 2^n - 1$, and give $M(8)$. **[3]**

Q39. Choosing data structures. Choose ONE structure and justify it with the cost. (a) The undo button of a text editor. (b) A playlist where songs are often inserted or removed in the middle once the position is known and no index access is needed. (c) A task manager that must always remove the smallest deadline first. **[3]**

Q40. Quicksort worst case. How many comparisons does Lomuto quicksort (last item as pivot) make on the sorted list `[1, 2, ..., 10]`? Give the formula, and say how a random pivot helps. **[3]**

— END OF PRACTICE SET —