CS 201: Data Structures & Algorithms

Quiz 1 — Complexity & Sorting — ANSWER KEY

Duration: 30 minutes Maximum marks: 13

SET B

Instructions

- Write your name and roll number before starting.
- This paper has 7 questions in 3 parts, for a total of 13 marks.

MCQ Answer Grid

Q	1	2	3	4
Ans	**A**	**D**	**B**	**B**

Part A — Multiple Choice

7 marks

Q1. Dijkstra's algorithm produces correct shortest paths on graphs with negative edge weights. *[1 mark]*

(A) False ✓
(B) True

> **Explanation.** A negative edge can improve an already-finalized vertex; use Bellman–Ford instead.

Q2. What does this function print for `n = 6`? *[2 marks]*

```
def f(n):
    c = 0
    while n > 1:
        n = n // 2
        c += 1
    print(c)
```

(A) `6`
(B) `1`
(C) `3`
(D) `2` ✓

> **Explanation.** $6 \to 3 \to 1$: two halvings.

Q3. By the master theorem, $T(n) = 2T(n/2) + n$ solves to: *[2 marks]*

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

Q4. What is the worst-case running time of binary search on a sorted array of n elements? *[2 marks]*

(A) $\Theta(n)$
(B) $\Theta(\log n)$ ✓
(C) $\Theta(n \log n)$
(D) $\Theta(1)$

Part B — Fill in the Blanks

2 marks

Q5. The average-case running time of quicksort with uniformly random pivots is $\Theta(\underline{\ n \log n\ })$. *[1 mark]*

Q6. A sorting algorithm is **stable** if it preserves the relative order of **equal** keys. *[1 mark]*

Part C — Short Answer *4 marks*

Q7. Sketch the decision-tree argument that any comparison-based sorting algorithm needs $\Omega(n \log n)$ comparisons in the worst case. *[4 marks]*

Model answer. Any comparison sort is a binary decision tree whose leaves are the $n!$ input permutations. A binary tree with $n!$ leaves has height at least $\log_2 n! = \Theta(n \log n)$ (Stirling), so some input forces that many comparisons.

Rubric. $+1$ model: binary decision tree, leaves = permutations; $+1$ at least $n!$ leaves; $+1$ height $\geq \log_2 n!$; $+1$ Stirling $\rightarrow \Omega(n \log n)$.

— End of paper —