National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Design and Analysis of Algorithms	Course Code:	
Program:	BS(SE)	Version:	6
Duration:	10 minutes	Total Marks:	12
Paper Date:	20 Mar, 2025	Weight:	
Section:	SE-6A	Pages:	02
Exam Type:	Quiz3-Version1		

8/12

Student Name: [redacted] Roll No #: [redacted] Section: [redacted]

Instructions: Do Rough Work on the right column.

1) In Counting Sort, which of the following is incorrect regarding its implementation?
a) It does not involve comparisons between elements
b) It modifies the input array in place
c) It requires an auxiliary array of size O(k)
d) It is stable when implemented correctly

✓

2) Why is Radix Sort not commonly used for general-purpose sorting?
a) It is asymptotically slower than Quick Sort
b) It requires additional space and is inefficient for large data
c) It is not stable
d) It does not work for integer sorting

O(d(n+k))
✓

3) If an array contains n elements drawn from a uniform distribution, what is the best way to determine the number of buckets?
a) Use n buckets
b) Use $\sqrt{n}$ buckets
c) Use log(n) buckets
d) Use k buckets

$\frac{100}{10}$ = 10 buckets
✓

4) Bucket Sort is generally inefficient when applied to:
a) Floating-point numbers uniformly distributed between [0,1]
b) Large datasets with arbitrary distributions
c) Sorting 32-bit integers in a small range
d) Sorting data with a normal distribution

O(n²)
✗

5) If $k \gg n$ (i.e., k is much larger than n), what is the primary issue when applying Counting Sort?
a) It still runs in O(n) time
b) Excessive memory usage
c) Requires additional recursive calls
d) Counting Sort is independent of k

✓

6) What is the worst-case time complexity of Radix Sort for sorting n integers when each integer has d digits and the stable subroutine (Counting Sort) takes O(n + b) time (with b being the base)?
a) O(n)
b) O(d · (n + b))
c) O(n log n)
d) O(n^2)

O(d(n+b))
b
✓

7) Suppose you need to sort an array of strings where each string consists of only lowercase English letters (a-z). What is the most efficient way to use Counting Sort?

a) Sort the strings character by character from right to left using Counting Sort
b) Use Counting Sort directly on the entire string
c) Convert each string into an integer and then use Counting Sort
d) Counting Sort cannot be used for sorting strings

8) You are given an array with n elements, where each element is in the range [0, k]. If $k = O(n^2)$, what is the time complexity of Counting Sort?

a) $O(n \log n)$
b) $O(n^2)$
c) $O(n + k)$
d) $O(n^3)$

9) Why must the subroutine in Radix Sort be stable?

a) To handle negative numbers
b) To preserve previous digit orderings
c) To reduce space complexity
d) To improve speed

10) How can Count Sort handle negative integers?

a) Ignore the negative sign
b) Split into positive/negative parts
c) Shift values to make them non-negative
d) Use a hash table

11) A bank needs to sort transactions by 3-digit branch codes (0-999). Which algorithm is best?

a) Radix Sort
b) Bucket Sort
c) Count Sort
d) QuickSort

12) Bucket Sort is generally inefficient when applied to:

a) Floating-point numbers uniformly distributed between [0,1]
b) Large datasets with arbitrary distributions
c) Sorting 32-bit integers in a small range
d) Sorting data with a normal distribution