Quiz 2. COL106 Sem-I, 2024-25 Paper Set-3
Wednesday October 23, 6:30 pm - 7:30 pm (20 points)

Name: Sourav Kumar Patel

Entry Number: 2023CS10751 Group: 04

1. (2 points) Suppose we use the hash function $f(x) = x \ mod \ 7$ in the following hash table of size 7. Use double hashing function $hash_2(x) = 5 - x \ mod \ 5$ to insert 76. 93, 40, 47, 33, 75 in the table in this order. Write down the entries in the table below after these insertions.

0	1	2	3	4	5	6
33	47	93	75		40	76

2. (2 points) Consider hashing with linear probing to insert: 11, 121, 33, 75, 4, 93, 100, 1 in a hash table of size 11, with the hash function $h(x) = x \ mod \ 11$. We now delete 33 from the table. Now, suppose you are searching for the key 2 in the resulting table. How many array indices (including blanks) will be accessed during this search operation?
Answer: 6

3. (2 points) You have a six-faced dice with numbers 1 to 6, in which the probability of getting an odd number is double the probability of getting an even number. All odd numbers are equally likely, and all even numbers are equally likely. You roll this dice three times, and add the numbers that you get in each roll. What is the expected value of this sum? Write your answer as a final number. Answer: 10

4. (2 points) Which of these statements is correct regarding the best-case time complexity of sorting algorithms? Here " > " represents (has asymptotically higher time complexity than). Here *SS*, *QS*, and *IS* represent Selection Sort, Quick Sort, and Insertion Sort respectively.
(a) $SS > QS > IS$ (b) $SS > IS > QS$ (c) $IS > QS > SS$
(d) $IS > SS > QS$ (e) $QS > SS > IS$ (f) $QS > IS > SS$ Answer: (a)

5. (2 points) Suppose we insert the elements (in this sequence) in a skip-list: 2, 5, 18, 10, 7 , 9. 10. Compute the probability that the number of levels in this skip list is at most 4. Note that the top level which has $+\infty$ and $-\infty$ is also counted. You can leave your answer in the form of a mathematical expression involving numbers only. Answer: $\left(\frac{7}{8}\right)^7$

6. (2 points) Consider the in-place implementation of QuickSort to sort numbers in ascending order. Assume that we always use the first element as the pivot. Let t1 and t2 be the number of comparisons by QuickSort for the inputs $\{1, 2, 3, 4, 5, 6, 7\}$ and $\{4, 2, 3, 1, 6, 5, 7\}$ respectively. Which one of the following holds?
(a) $t1 = t2$ (b) $t1 < t2$
(c) $t1 > t2$ (d) can't say (depends on other aspects of implementation)
Answer: (C)

7. (2 points) Given a binary string $s = s_0 \ s_1 \ ... \ s_{n-1}$ we compute its hashmap as follows: $h(s) = (\overset{1}{s_0} + \overset{0}{s_1}z + \overset{0}{s_2}z^2 + ... + s_{n-1}z^{n-1}) \ mod \ 7$, where z = 11. What will be the hash-map of the string 10001001? You should write the final answer as a simplified number.
Answer: 2

8. (2 points) Consider the following pseudo-code for merge-sort in ascending order which also has a print statement inside it.

```
mergesort(A) {
    Let n be the length of array A
    (i.e., the number of elements in A)
    If  (n <= 1) return A
    Else
        A1 = mergesort(Left half of A (i.e., A[0]...A[n/2-1]))
        A2 = mergesort(right half of A (i.e, A[n/2]...A[n-1]))
        B = Merge(A1, A2) (The Merge procedure returns an array)
        For each element in B (Iterate left to right)
            Print element(no newline is introduced after printing)
        Print ** (no newline is introduced after printing)
        Return B
}
```

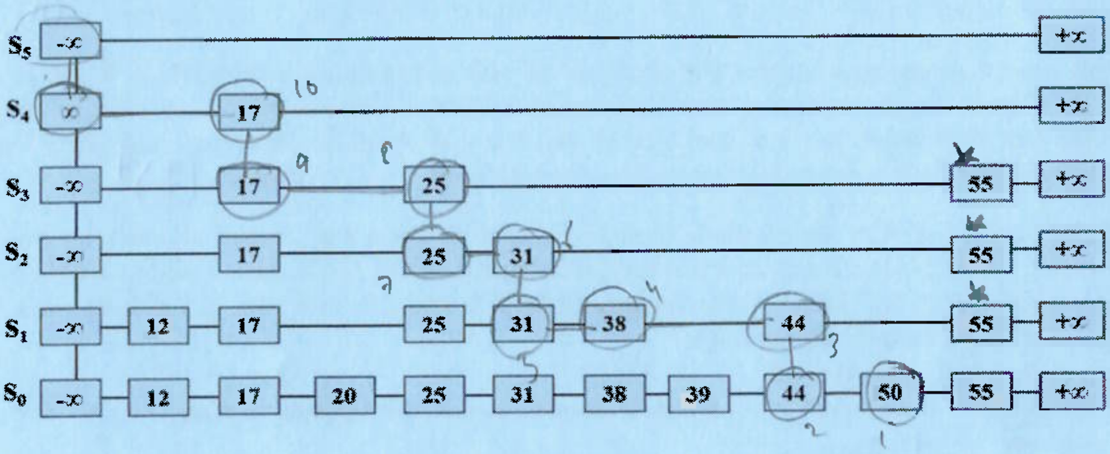
Show the output when run on the array: [47, 25, 74, 12, 15, 90, 40, 20].

Answer: 2547** 1274** 12254774** 1590** 2040** 15204090** 1215202540457490**

9. (2 points) We're debugging a quicksort implementation that is supposed to sort an array in ascending order. After the first partition step has been completed, the contents of the array are in the following order: 3, 9, 1, 14, 24, 22, 27, 30. Which of the numbers in the array could have been a possible pivot? (List all such numbers)

Answer: 14, 27, 30

10. (2 points) Consider the following skip-list.



How many skip list elements, aggregated over all the levels, are accessed (excluding those representing +∞ and −∞) while searching for 50? Answer: ~~10~~ 13

[we go through 10 nodes but to check whether our element is < next node we ~~de~~ also acess the next node. 3 extra nodes excluding -∞, ∞]