Summary in Graph

Exam Summary (GO Classes CS Test Series | Algorithms | Topic Wise Test 1)

10.7% 2.67/25

3 Correct | 3 Wrong | 9 Skipped | 34m of 45m

Qs. Attempted:	6 (3 + 3)	Correct Marks:	4 (2 + 2)	Total Questions:	15 (5 + 10)
Correct Attempts:	3 (2 + 1)	Penalty Marks:	1.33 (0 + 1.33)	Total Marks:	25 (5 + 20)
Incorrect Attempts:	3 (1 + 2)	Resultant Marks:	2.67 (2 + 0.67)	Exam Duration:	45 Minutes
				Time Taken:	34 Minutes

Individual Question Statistics

View Order: Sorted Order | Select Metric: Marks Obtained



* Average Stats based on first attempt stats of Top 10% (84) out of 845 users

EXAM RESPONSE | EXAM STATS | FEEDBACK

Technical

Q #1 Multiple Select Type | Award: 1 | Penalty: 0 | Algorithms | Medium (41%)

A lazy programmer thinks that implementing the procedure Merge (of the merge sort) is too complicated. To simplify merge sort he eliminates the merge all together, and only recursively "sorts" the two halves of the array. Here is his pseudo-code:

```
procedure LAZYSORT(A[1...n])
    if n>1 then
        n =n/2
        LAZYSORT(A[1...n])
        LAZYSORT(A[n+1...n])
```

Which of the following is/are TRUE about LAZYSORT?

A. The Time Complexity of LAZYSORT is $\Theta(n \log n)$
B. The Time Complexity of LAZYSORT is $\Theta(n)$
C. LAZYSORT changes the original array in such a way that it exchanges the first half of the array with the second half.
D. LAZYSORT does not change array elements at all

Your Answer: Correct Answer: B,D | Not Attempted | Time taken: 0 min 20 sec | Discuss

Q #2 Multiple Choice Type | Award: 2 | Penalty: 0.33 | Algorithms | Easy (77%)

3-Way-Merge Sort: Suppose that instead of dividing in half at each step of Merge Sort, you divide into thirds, sort each third, and finally combine all of them using a three-way merge subroutine. What is the overall asymptotic running time of this algorithm? (Hint: Note that the merge step can still be implemented in $O(n)$ time.)

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

Your Answer: B | Correct Answer: B | Correct | Time taken: 0 min 51 sec | Discuss

Q #3 Multiple Select Type | Award: 1 | Penalty: 0 | Algorithms | Medium (59%)

Recall the Partition subroutine that we used in QuickSort. Suppose that the following array has just been partitioned around some pivot element
: 3, 1, 2, 4, 5, 8, 7, 6, 9.

Which of these element(s) could have been the pivot element?

A. 4
B. 5
C. 2
D. 9

Your Answer: D | Correct Answer: A,B,D | Incorrect | Time taken: 0 min 34 sec | Discuss

Q #4 Multiple Choice Type | Award: 1 | Penalty: 0.33 | Algorithms | Easy (78%)

Let $T(n) = 47T(n/3) + n^{\log_3 4}$. What will be asymptotic bound on $T(n)$?

A. $T(n) = \Theta\left(n^{\log_3 4} \log n\right)$
B. $T(n) = \Theta(n \log n)$
C. $T(n) = \Theta\left(4^{\log_3 n} \log \log n\right)$
D. None of these

Your Answer: A | Correct Answer: A | Correct | Time taken: 0 min 27 sec | Discuss

Q #5 Multiple Choice Type | Award: 1 | Penalty: 0.33 | Algorithms | Medium (45%)

Insertion sort performs best case when input is sorted and worst case when input is sorted backwards.
Compute the average COST of insertion sort to sort a vector with n different integers when with probability $\frac{99.5}{n}$ one chooses a vector which is sorted backwards and with probability $1 - \frac{99.5}{n}$ one chooses a sorted vector.

A. $\Theta\left(n^2\right)$
B. $\Theta(n \log n)$
C. $\Theta(n)$
D. $\Theta\left(n^3\right)$

Your Answer: Correct Answer: B | Not Attempted | Time taken: 0 min 32 sec | Discuss

Q #6 Numerical Type | Award: 2 | Penalty: 0 | Algorithms | Hard (26%)

Consider a constant $\alpha = 0.3$. If we use randomized quick sort (pivot chosen is uniform random) then what is the probability that the partition method will result in an array such that the size of the smaller subarray is $\geq \alpha n$ where n is the total size of the array?

Your Answer: Correct Answer: 0.4 | Not Attempted | Time taken: 0 min 17 sec | Discuss

Q #7 Multiple Select Type | Award: 2 | Penalty: 0 | Algorithms | Medium (71%)

Consider a quicksort where splits at every level are in the proportion of α, which means one of the partitions is αn and the other partition is $(1-\alpha)n$. The recursive equation of time complexity can be written as

$$T(n) = T(\alpha n) + T((1-\alpha)n) + n.$$

Consider a recursion tree of function calls and let x be the height of the tree when we hit the base case the first time and let y be the maximum height of the tree then, which of the following is/are true?
Base case is an array containing single element.

A. $\alpha^x n = 1$ and $(1-\alpha)^y n = 1$ if $\alpha < 0.5$
B. $\alpha^y n = 1$ and $(1-\alpha)^x n = 1$ if $\alpha < 0.5$
C. $\alpha^x n = 1$ and $(1-\alpha)^y n = 1$ if $\alpha > 0.5$
D. $\alpha^y n = 1$ and $(1-\alpha)^x n = 1$ if $\alpha > 0.5$

Your Answer: Correct Answer: A,D | Not Attempted | Time taken: 0 min 0 sec | Discuss

Q #8 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (55%)

Consider partition algorithm of randomized quick sort which randomly choose pivot element and split the array into two halves accordingly. Pivot element will not be in any of the subarrays.

Suppose array length n is which is odd then what is the probability that median element lies in second half after partition?

A. $1/2$
B. $1/2 - 1/2n$
C. $1/2 - 2/n$
D. $1 - 1/2n$

Your Answer: Correct Answer: B | Not Attempted | Time taken: 0 min 1 sec | Discuss

Q #9 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (59%)

How many calls of merge-sort are performed when sorting an array of size 64? Base case is an array containing a single element.

A. 2^{63}
B. 127
C. 63
D. 64

Your Answer: Correct Answer: B | Not Attempted | Time taken: 0 min 0 sec | Discuss

Q #10 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (58%)

Suppose you are given k sorted arrays, each with n elements, and you want to combine them into a single array of kn elements. Consider the following approach. Divide the k arrays into $k/2$ pairs of arrays, and use the Merge subroutine of MergeSort to combine each pair. Now you are left with $k/2$ sorted arrays, each with $2n$ elements.

Repeat this approach until you have a single sorted array with kn elements. What is the running time of this procedure, as a function of k and n?

A. $\Theta(nk \log k)$
B. $\Theta\left(nk^2\right)$
C. $\Theta(nk \log n)$
D. $\Theta(n \log k)$

Your Answer: C | Correct Answer: A | Incorrect | Time taken: 0 min 5 sec | Discuss

Q #11 Multiple Select Type | Award: 2 | Penalty: 0 | Algorithms | Easy (81%)

Consider an array of n distinct integers. An array element is a peak if it is greater than its neighbors.
For corner elements, we need to consider only one neighbor. The corner element is peak only when it's greater than its neighbors.
If there are more than one peak item, simply return one of them.

Example:

- Input : [1, 5, 3, 2, 4, 0] Output : 5
- Input : [1, 2, 3, 4, 5, 6] Output : 6
- Input : [7, 6, 5, 4, 3, 2] Output : 7

We design a divide-and conquer algorithm that solves this problem. The incomplete pseudo-code is given below.

```
findPeak(A, 1, n)
    if(1==n) //one element
        return 1
    if(A==1 ) //two elements
        if(A[1]> A[2] ) return 1
        else return 2
    mid = (1+n)/2
    if (A[mid]> A[mid-1] and A[mid]> A[mid+1]) return mid
    if (A[mid-1]> A[mid])
    //line X
    if (A[mid+1]> A[mid])
    //line Y
}
```

Fill in the blanks for lines X and Y.

A. Line X is return findPeak $(A, i, \text{mid}-1)$
B. Line Y is return findPeak A, mid $+1, j)$
C. Line X is return findPeak $(A$ mid $+1, j)$
D. Line Y is return findPeak $(A, i, \text{mid}-1)$

Your Answer: Correct Answer: A,B | Not Attempted | Time taken: 0 min 0 sec | Discuss

Q #12 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Hard (0%)

Consider two statements about RANDOMIZED QUICKSORT.

- $S1$: There is some input for which randomized quicksort always runs in time $\Theta\left(n^2\right)$ time.
- $S2$: An adversary can provide randomized quicksort with an input array of length n that forces the algorithm to run in $\Theta(n \log n)$ time on that input.

Choose the correct option (assume distinct elements in input):

A. Both statements are correct
B. $S1$ is correct, but $S2$ is false
C. $S1$ is false, but $S2$ is correct
D. Both Statements are incorrect

Your Answer: Correct Answer: D | Not Attempted | Time taken: 0 min 1 sec | Discuss

Q #13 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (58%)

An inversion in a vector of integers $T[0 \ldots n-1]$ is a pair of positions storing values that are not ordered, that is, a pair (i, j) such that $0 \leq i < j < n$ and $T[i] > T[j]$.

Let $T[0 \ldots 99]$ be an array of hundred integers. It is also given that $T[6] > T[10]$. Relation between other elements is not given.
Which of the following is/are true about a number of inversions.

A. There are at most 11 inversions in the array.
B. There are at least 10 inversions in the array.
C. There are at most 10 inversions in the array.
D. There are at least 11 inversions in the array.

Your Answer: B | Correct Answer: B | Correct | Time taken: 0 min 5 sec | Discuss

Q #14 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (47%)

Assume that the inputs of size n of a certain algorithm are of the following types:

- Type 1 : for each input of this type, the algorithm takes time $\Theta\left(n^3\right)$. Moreover, the probability that the input is of this type is $\frac{1}{2}$.
- Type 2 : for each input of this type, the algorithm takes time $\Theta\left(n^2\right)$. Moreover, the probability that the input is of this type is $\frac{1}{2}$.
- Type 3 : for each input of this type, the algorithm takes time $\Theta(n)$. Moreover, the probability that the input is of this type is $1 - \frac{1}{2^n} - \frac{1}{n}$.

Then the cost of the algorithm in the average case is

A. $\Theta\left(n^3\right)$
B. $\Theta\left(n^2\right)$
C. $\Theta(n)$
D. $\Theta\left(n^2\right)$

Your Answer: Correct Answer: D | Not Attempted | Time taken: 0 min 0 sec | Discuss

Q #15 Multiple Choice Type | Award: 2 | Penalty: 0.67 | Algorithms | Medium (37%)

Let $A[1 \ldots n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

Suppose that each element of A is chosen randomly, independently, and uniformly from the range 1 through n.

What will be the expected number of inversions?

A. $n(n-1)/4$
B. n
C. $n(n-1)/2$
D. $n^2/2$

Your Answer: C | Correct Answer: A | Incorrect | Time taken: 0 min 4 sec | Discuss

Copyright & Stuff