Published Preview Edit

Week 06 Quiz

This is a preview of the draft version of the quiz.

By submitting work for this quiz I hereby declare that I understand the University's policy on academic integrity (https://academicintegrity.unimelb.edu.au/) and that the work submitted is original and solely my work, and that I have not been assisted by any other person (collusion) apart from where the submitted work is for a designated collaborative task, in which case the individual contributions are indicated. I also declare that I have not used any sources without proper acknowledgment (plagiarism). Where the submitted work is a computer program or code, I further declare that any copied code is declared in comments identifying the source at the start of the program or in a header file, that comments inline identify the start and end of the copied code, and that any modifications to code sources elsewhere are commented upon

Quiz Type	Graded Quiz
Points	7
Assignment Group	Assignments
Shuffle Answers	Yes
Time Limit	No Time Limit
Multiple Attempts	Yes
Score to Keep	Highest
Attempts	Unlimited
View Responses	Always
Show Correct Answers	No
One Question at a Time	No

Due	For	Available from	Until
-	Everyone	-	-

Preview

Correct answers are hidden.

Score for this attempt: **7** out of 7

Submitted Sep 11 at 14:34

This attempt took less than 1 minute.

Question 1

1 / 1 pts

The recurrence relation for merge sort is:

- T(n) = 2T(n/2) + O(1)
- T(n) = T(n/2) + O(1)
- T(n) = T(n-1) + O(n)
- T(n) = 2T(n/2) + O(n)

Let's think about how merge sort works. We split our array or list in to two halves, and run merge sort recursively on each half (this is where the 2T(n/2) part arises). We then merge the two sorted halves (this is where the O(n) part arises).

Question 2

1 / 1 pts

Quicksort uses Hoare partitioning. Assume an array contains ten keys: 6 3 1 7 9 5 8 2 4 0. After a first round of simple Hoare partitioning (not median-of-three), the array looks like so:

- 5 3 1 0 4 2 6 8 9 7
- 2 3 1 0 4 5 6 8 9 7
- 2 3 0 1 5 4 6 7 8 9
- 5 3 1 4 0 2 6 7 9 8

○ 3 1 5 2 4 0 6 7 9 8

Well done!

Question 3

1 / 1 pts

Consider this recurrence relation:

T(1) = 1
T(n) = 2 T(n/3) + 2n + 1   for n>1

The Master Theorem says that

○ $T(n)\ \epsilon\, \Theta\left(n^2\right)$

◉ $T(n)\ \epsilon\, \Theta(n)$

○ $T(n)\ \epsilon\, \Theta(n\ \log\ \log\ n)$

○ $T(n)\ \epsilon\, \Theta\left(n^3\right)$

○ $T(n)\ \epsilon\, \Theta(n\ \log\ n)$

That's right. In this case we have a=2, b=3, and d=1. And indeed 2 < 3.

Question 4

1 / 1 pts

Consider this recurrence relation:

T(1) = 1

T(2) = 1
T(n) = 4 T(n-2) + 2n^2 for n>2

The Master Theorem tells us

- $T(n) \; \epsilon \; \Theta \left(n^3\right)$
- $T(n) \; \epsilon \; \Theta \left(n^2 \; \log \; n\right)$
- $T(n) \; \epsilon \; \Theta \left(n \; \log \; n\right)$
- nothing
- $T(n) \; \epsilon \; \Theta \left(n^2\right)$

That's right, the Master Theorem does not help here, as the recurrence is not of the required form.

Question 5

1 / 1 pts

Which of the following sorting algorithm has the running time that is least dependant on the initial ordering of the input?

- Merge sort
- Insertion sort
- Selection sort
- Quick sort

Well done

Question 6

1 / 1 pts

Suppose we have an array A with 33,554,431 elements. We want to apply binary search to look for some element k. A test of the form "is k = A[i]?" is a probe. How many probes will be performed in the worst case?

25

Yes, the number of elements is 2^{25} - 1. We have a worst-case instance if k is not in the array.

Question 7

1 / 1 pts

What is the postorder traversal sequence for a binary tree whose preorder traversal sequence is A, B, C, D, E, F, G, H, I and whose inorder sequence is C, B, E, D, F, A, G, I, H ?

- (selected) None of the above
- C, E, F, B, D, I, H, G, A
- C, E, F, D, B, H, G, I, A
- C, E, F, D, B, H, I, G, A
- C, E, F, B, D, H, I, G, A

That's correct. In fact the postorder sequence is C, E, F, D, B, I, H, G, A.

Quiz Score: **7** out of 7