Week 06 Quiz

Plagiarism declaration

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 as to the nature of the modification.

Due Apr 23 at 23:59 **Points** 10 **Questions** 9
Available Apr 14 at 10:00 - Apr 23 at 23:59 10 days **Time Limit** None
Allowed Attempts Unlimited

Instructions

You should attempt the quiz after the lecture and your tutorial.

- You may attempt the quiz multiple times (if you happen to get a question wrong, you can do it again)
- Your score on the quiz will be recorded in the grade book.
- The quiz might not display equations correctly in some browsers. If you experience problems, we recommend that you use Firefox.

Note: you must complete at least eight of the weekly quizzes to meet one of the hurdle requirements in this subject

This quiz was locked Apr 23 at 23:59.

Attempt History

Attempt	Time	Score

	Attempt	Time	Score
LATEST	Attempt 1	1,280 minutes	8 out of 10

Score for this attempt: **8** out of 10
Submitted Apr 23 at 21:08
This attempt took 1,280 minutes.

Question 1

1 / 1 pts

The recurrence relation for merge sort is:

Correct!

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

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 4 0 2 6 7 9 8

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

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

- $T(n) \in \Theta(n^3)$
- $T(n) \in \Theta(n^2)$
- $T(n) \in \Theta(n \log \log n)$
- Correct! $T(n) \in \Theta(n)$
- $T(n) \in \Theta(n \log n)$

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

- $T(n) \in \Theta(n \log n)$
- $T(n) \in \Theta(n^3)$
- $T(n) \in \Theta(n^2 \log n)$
- $T(n) \in \Theta(n^2)$

Correct!

- nothing

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

Question 5

0 / 1 pts

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

orrect Answer

- Selection sort

ou Answered

- Merge sort

- Quick sort
- Insertion sort

Try again (see lecture slides)

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?

Correct!

25

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

orrect Answers

25 (with margin: 0)

Question 7

1 / 1 pts

What is the tight bound using Big-Theta notation for the time complexity of the following recurrence relation:

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

when the Master Theorem is used?

- $\theta(n \log n)$

- $\theta\left(n^{\log_3 2}\right)$

Correct!

- $\theta\left(n^{\log_2 3}\right)$
- $\theta\left(\log_2 n\right) + O(1)$

Question 8

2 / 2 pts

What is the tight bound using Big-Theta notation for the time complexity of the following recurrence relation:

$T(n) = 3T\left(\frac{n}{9}\right) + \sqrt{n}$

when the Master Theorem is used?

Correct!

- $\theta\left(n^{\frac{1}{2}} \log n\right)$
- $\theta\left(\sqrt{n}\right)$
- $\theta\left(n^3\right)$
- $\theta\left(\log n^3\right)$

Question 9

0 / 1 pts

Consider a modification to QuickSort where each time the partition function is called, the median of the partition array is always found (in constant time) and used as the pivot.

The worst-case running time for the algorithm is:

orrect Answer

- $\Theta\left(n \log n\right)$

$\Theta(n)$

ou Answered

$\Theta(\log\ n)$

$\Theta(n^2)$

Quiz Score: **8** out of 10