Module 4 Graded Quiz

Due Oct 8, 2021 at 11:59pm **Points** 13 **Questions** 13
Available Sep 25, 2021 at 12am - Feb 8 at 11:59pm 5 months
Time Limit 120 Minutes

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	60 minutes	13 out of 13

Score for this quiz: **13** out of 13
Submitted Oct 8, 2021 at 4:30pm
This attempt took 60 minutes.

Question 1

1 / 1 pts

Solve the following recurrence relation using any method. Provide your answer in big-O notation:

$T(n) = 2T\left(\frac{n}{2}\right) + \log(n)$ for n > 1, 1 otherwise

- T(n) = O(logn)
- T(n) = O(nlogn)
- T(n) = O(n^2)
- Correct! T(n) = O(n)

Question 2

1 / 1 pts

Suppose we have a modified version of Merge-Sort that at each recursive level splits the array into two parts each of size ¼ and ¾ respectively. Also, assume the size of any given input array is a power of four. Give the asymptotic time complexity of this Merge-Sort variant.

- O(n)

Correct!

- O(nlogn)
- O(logn)
- O(n^2)

Question 3

1 / 1 pts

Suppose we modify the combine step of the closest pair of points algorithm such that distance δ from dividing line *L* is updated immediately to δ' whenever the distance between two points on either side of *L* is discovered to be less than δ. In this sense, we allow δ to assume multiple values during the same combine step for each recursive call. Determine whether the following statement is true or false and explain your reasoning: The time complexity of the closest pair of points algorithm is guaranteed to be improved by a constant factor.

- True: If δ is reduced every time a new closest pair of points is found during the combine step, then it will sometimes be the case that a fewer number of future comparisons will be necessary after d is adjusted to δ'.

Correct!

- False: If δ is reduced only after the last pair of points sorted by y-coordinate within δ of L is compared, then no additional benefit will be gained.

True: If δ is reduced every time a new closest pair of points is found during the combine step, then it will always be the case that a fewer number of future comparisons will be necessary after d is adjusted to δ'.

False: The number of comparisons made between points within δ of L would necessarily be the same as without the modification.

Question 4

1 / 1 pts

Suppose there is a set of *n* points each with the same *x*-coordinate. Obviously, this will cause the closest pair of points algorithm to fail. Since this is the case, what must be the smallest possible asymptotic time complexity required to find the closest pair of points in this instance?

- $O(n\log^2 n)$
- $O(n^2)$
- Correct! $O(n\log n)$
- $O(n)$

Question 5

1 / 1 pts

Given a two-dimensional plane with points (0,1),(1.5,2),(1,4),(4.8,2),(5,3),(7,3.5),(8,8),(7.5,9.5), give the numerical value of δ *before* the combine step on the highest level of the recursive stack.

- 1.80

- 2.06

Correct!

- 1.58
- 1.02

Question 6

1 / 1 pts

Given a two-dimensional plane with points (1.5,1),(2,3.8),(2.75,1),(5,3.3),(6,3.5),(7.5,2),(8.5,1.2),(8,5), give the numerical value of δ *after* the combine step on the highest level of the recursive stack.

Correct!

- 1.02
- 0.89
- 1.28
- 1.25

Question 7

1 / 1 pts

Identify which of the following statements is true regarding the Karatsuba multiplication algorithm. Let *n* be the number of bits used to represent both the multiplicand and multiplier.

i.) The Karatsuba algorithm is asymptotically less complex than merge-sort with respect to the number of operations it requires to terminate.

ii.) Each call to this algorithm produces three additional recursive calls on roughly half the number of bits as the previous call.

- Neither of these are true

i. only

Correct!

ii. only

i. and ii.

Question 8

1 / 1 pts

Suppose Anatolii Karatsuba had somehow discovered a way to produce the product of integers *x* and *y* with only two real multiplications as opposed to 3 (as seen in the Karatsuba algorithm presented in class). Suppose also that this new algorithm required an asymptotically linear number of adding, subtracting and shifting operations per recursive call to his algorithm. What would have been the asymptotic time complexity of his algorithm if *x* and *y* are each represented by *n* bits?

(Hint: Read the supplementary document about non-binary recursive trees carefully before answering.)

$O(n^{1.585})$

O(n)

Correct!

O(nlogn)

$O(n^2)$

Question 9

1 / 1 pts

Identify which of the following statements are true regarding Strassen's fast matrix multiplication algorithm as applied to matrices of size *n* by *n*.

i.) Strassen's algorithm is the fastest known algorithm for any type of matrix multiplication.

ii.) If *n* is not a power of 2, adding rows and columns of zero-entries until *n* is a power of 2 will allow the algorithm to run normally and produce the correct result.

- i. only
- Neither of these are true
- i. and ii.

Correct!

- ii. only

Question 10

1 / 1 pts

Suppose Volker Strassen had somehow discovered a way to produce the product of two *n* by *n* matrices with only six real multiplications and an asymptotically quadratic number of addition and subtraction operations per recursive call to his algorithm. What would have been the asymptotic time complexity of his algorithm?

(Hint: Read the supplementary document about non-binary recursive trees carefully before answering.)

Correct!

- $O(n^{2.585})$
- $O(n^2 \log n)$
- $O(n^{2.810})$
- $O(n^{2.322})$

Question 11

1 / 1 pts

We propose the following divide-and-conquer algorithm for solving the minimum spanning tree problem for a connected, weighted graph $G = (V, E)$. Recursively, we divide the set of vertices in roughly half to obtain vertex sets V_1, V_2. We recursively compute the minimum spanning trees T_1, T_2 over V_1 and V_2, respectively. To combine the subproblems, we add the smallest edge such that T_1, T_2 now form a single component. What is the primary difficulty of achieving an efficient implementation under this framework?

- ( ) The conquer step: there may be an exponential number of recursive calls per level.
- ( ) The base case of the recursion will require traversing a recursive depth exponential in the size of V.
- ( ) The combine step: searching for an edge of minimal weight to join the two trees will require exponential time
- Correct! (•) The divide step: Partitioning the vertices in two equal-sized components such that each component is connected may require considering all vertex subsets of size $n/2$.

Question 12

1 / 1 pts

Consider the following divide-and-conquer algorithm for checking to see whether an element e exists in a pre-sorted array A.

Divide A into two roughly equal sized subarrays A_1, A_2 at element e'. If $e \leq e'$, recursively check to see if e is in A_1. Otherwise, recursively check to see if e is in A_2. Return true if so, and false otherwise. The base case considers an array of size one, where it is trivial to check whether e is the singleton element.

Give the recurrence relation that describes the algorithm above in terms of the number n of elements in A. Assume $T(n) = 1$ when $n = 1$.

Correct!

- $T(n) = T\left(\frac{n}{2}\right) + O(1)$
- $T(n) = T\left(\frac{n}{2}\right) + O(n)$
- $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$
- $T(n) = 2T\left(\frac{n}{2}\right) + O(1)$

Question 13

1 / 1 pts

Consider the following divide-and-conquer algorithm used to find the maximum element of an unsorted array A:

Divide A into two roughly equal parts A_1 and A_2. Recursively find the maximum elements e_1, e_2 of their respective subarrays. Return maximum of e_1, e_2.

Identify the asymptotic running time of the *combine step* in terms of the number n of elements in A.

- $\Theta(\log n)$

Correct!

- $\Theta(1)$
- $\Theta(n)$
- $\Theta(n \log n)$

Quiz Score: **13** out of 13