X

(https://swayam.gov.in) (https://swayam.gov.in/nc_details/NPTEL)

reviewer4@nptel.iitm.ac.in ⌄

NPTEL (https://swayam.gov.in/explorer?ncCode=NPTEL) » Design and analysis of algorithms (course)

Announcements (announcements) **About the Course (https://swayam.gov.in/nd1_noc20_cs27/preview)**

Ask a Question (forum) Progress (student/home) Mentor (student/mentor)

Unit 16 - Week 5 Quiz

Course outline

Week 5 Quiz

The due date for submitting this assignment has passed. **Due on 2020-03-04, 23:59 IST.**
As per our records you have not submitted this assignment.

All questions carry equal weightage. You may submit as many times as you like within the deadline. Your final submission will be graded.

1) An image is represented as an N×N array A of pixels. There is a function that transforms the image as follows: ***2 points***

- The function scans each pixel `A[i][j]`. Depending on the current value of `A[i][j]`, some values in row `i` and column `j` are updated. It is possible that all values in row `i` and column `j` are updated. It is also possible that no values are updated.
- Updating a pixel takes time O(log N).
- Each pixel in the image is updated at most once by the function.

What is the best upper bound you can estimate for the total time taken across all the updates made by the function?

○ $O(N^3 \log N)$

○ $O(N^3)$

○ $O(N^2 \log N)$

○ $O(N^2)$

No, the answer is incorrect.
Score: 0

Feedback:

A naive analysis says that we update upto O(N) entries when scanning each `A[i][j]`. Thus the update at each `A[i][j]` takes O(N log N) in the worst case, so $O(N^3 \log N)$ overall, since there are N^2 entries in the array.

However, since each of the N^2 entries is updated at most once, an amortized analysis says that the total time cannot exceed $O(N^2 \log N)$.

Accepted Answers:

Week 4 : Weighted graphs

Week 4 Quiz

Week 4 Programming Assignment

Week 5: Data Structures: Union-Find and Heaps

Week 5 : Divide and Conqure

Week 5 Quiz

○ **Quiz : Week 5 Quiz (assessment?name=113)**

Week 6: Data Structures: Search Trees

Week 6: Greedy Algorithms

Week 6 Quiz

Week 6 Programming Assignment

Week 7: Dynamic Programming

Week 7 Quiz

Week 7 Programming Assignment

Week 8: Linear Programming and Network Flows

Week 8: Intractability

Week 8 Quiz

$O(N^2 \log N)$

2) In the Union-Find data structure, suppose we want to add an operation `reassign(j,k)` that moves item `j` to the same component as item k. What would be the complexity of this operation for a set with n elements? ***2 points***

○ O(1) for both the array representation and the pointer representation.
○ O(1) for the array representation, but not O(1) for the pointer representation.
○ O(1) for the pointer representation, but not O(1) for the array representation.
○ Not O(1) for either the array representation or the pointer representation.

No, the answer is incorrect.
Score: 0
Feedback:
In the array representation, we have to assign Component[j] the same value as Component[k]. In the pointer representation, we look up Node[j] and Node[k] and copy the parent pointer from Node[k] to Node[j So both these updates take time O(1).

However, in the array representation we also have to update members[i] for both components, which will take time proportional to the sizes of the components, so it is not a constant time operation.

In the pointer representation we also have to shift the pointer of the children of j to the parent of j. Since j could have multiple children, this is again not a constant time operation.

Accepted Answers:
Not O(1) for either the array representation or the pointer representation.

3) In a **min-heap**, what is the most accurate description of the worst-case complexity of the operation `find_max` that reports the value of the largest element in the heap, without removing it? ***2 points***

○ O(1)
○ O(log n)
○ O(n)
○ O(n log n)

No, the answer is incorrect.
Score: 0
Feedback:
The min-heap structure does not help us find the maximum value in any way, so the simplest option is to scan the heap as an unsorted array and return the maximum value in O(n) time.

Accepted Answers:
O(n)

4) After inserting 57 into a max-heap, the structure of the heap is [82,57,27,42,25,18,25,27,32]. What was the structure of the heap before inserting 57? ***2 points***

○ [82,42,32,27,25,18,25,27]
○ [82,27,42,32,25,18,25,27]
○ [82,42,27,25,32,18,25,27]
○ [82,42,27,32,25,18,25,27]

No, the answer is incorrect.
Score: 0
Feedback:
The leaf node where 57 was inserted was the last one, which is currently 32. The path from the current position of 57 to this leaf goes via the node with value 42. So we have to undo the swap made along this path when inserting 57.

Accepted Answers:
[82,42,27,32,25,18,25,27]

5) Consider an alternative to binary search called ternary search that divides the input array into three equal parts and recursively searches one of these three segments. What can we say about the ***2 points***

Text Transcripts

Books

Download Videos

asymptotic worst case complexity of ternary search versus binary search?

○ The complexity of ternary search is the same as that of binary search.

○ The complexity of binary search is strictly better than that of ternary search.

○ The complexity of ternary search is strictly better than that of binary search.

○ The relative complexity of ternary and binary search depends on the distribution of values across the array.

No, the answer is incorrect.
Score: 0

Feedback:

The recurrence for ternary search is

- *$T(0) = T(1) = 1$*
- *$T(n) = O(1) + T(n/3)$*

which resolves as $T(n) = O(\log_3 n)$. However, since $\log_3 n = \log_2 n/\log_2 3$, the asymptotic complexity is the same as $O(\log_2 n)$ for binary search.

Accepted Answers:

The complexity of ternary search is the same as that of binary search.