Click to register for Certification exam

If already registered, click to check your payment status

Course outline

- About NPTEL
- How does an NPTEL online course work?
- Week 1 : Introduction
- Week 1 : Analysis of algorithms
- Week 1 Quiz
- Week 2 : Searching and sorting
- Week 2 Quiz
- Quiz: Week 2 Quiz
- Week 2 Programming Assignment
- Week 3 : Graphs
- Week 3 Quiz
- Week 3 Quiz
- Week 3 Programming Assignment
- Text Transcripts
- Books
- Download Videos
- Lecture Material
- Problem Solving Session - Jan 2024

Thank you for taking the Week 2 Quiz.

Week 2 Quiz

Your last recorded submission was on 2024-02-06, 16:43 IST **Due date: 2024-02-07, 23:59 IST.**

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

1) Arrays A and B each contain N integers arranged in a random sequence. We want to check if A and B have any entries in common. Which of the following would be the most efficient algorithm, asymptotically? ***2 points***

- For each pair of positions $0 \le i,j < N$, check if A[i] is equal to B[j].
- Sort A using merge sort. For each element B[i], try to insert B[i] in A and see if you encounter a duplicate.
- Sort A using quicksort. For each element B[i], use binary search to check if B[i] appears in A.
- **Sort A and B using merge sort. Merge A and B to check for duplicates.**

2) Suppose our aim is to sort an array in ascending order. Which of the following statements is true? ***2 points***

- **Input in descending order is worst case for both selection sort and insertion sort.**
- Input in descending order is worst case for selection sort but not for insertion sort.
- Input in ascending order is worst case for both selection sort and insertion sort.
- Input in ascending order is worst case for insertion sort but not for selection sort.

3) Suppose we want to sort an array in descending order and we implement quicksort so that we always choose the last element in the array as the pivot element. Assume that the input is a permutation of {1, 2, ..., n}. Which of the following would ***definitely*** be a worst case permutation of this input for this implementation of quicksort? ***2 points***

- {1, 2, ..., n} with all odd numbers in ascending order followed by all even numbers in random order.
- {1, 2, ..., n} with all even numbers in descending order followed by all odd numbers in ascending order.
- **{1, 2, ..., n} in descending order.**
- {1, 2, ..., n} in some random order.

4) Which of the following statements is ***not*** true? ***2 points***

4) Which of the following statements is ***not*** true? ***2 points***

- Quicksort and merge sort are both examples of divide and conquer algorithms.
- **If we randomly choose a pivot element each time, quicksort will always terminate in time O(n log n).**
- For every fixed strategy to choose a pivot for quicksort, we can construct a worst case input that requires time $O(n^2)$.
- If we could find the median in time O(n), quicksort would have worst case complexity O(n log n).

5) We have a list of pairs [("Ashwin",69),("Sumati",87),("Tanuja",69),("Brinda",87), ("Shabana",72),("Vijay",60)], where each pair consists of a student's name and his/her marks in a course. We sort these pairs in ascending order of marks. Which of the following corresponds to a stable sort of this input? ***2 points***

- [("Vijay",60),("Tanuja",69),("Ashwin",69),("Shabana",72),("Sumati",87),("Brinda",87)]
- **[("Vijay",60),("Ashwin",69),("Tanuja",69),("Shabana",72),("Sumati",87),("Brinda",87)]**
- [("Vijay",60),("Tanuja",69),("Ashwin",69),("Shabana",72),("Brinda",87),("Sumati",87)]
- [("Vijay",60),("Ashwin",69),("Tanuja",69),("Shabana",72),("Brinda",87),("Sumati",87)]

You may submit any number of times before the due date. The final submission will be considered for grading.

Submit Answers

Week 2 Quiz

Due date: 2024-02-07, 23:59 IST.

Your last recorded submission was on 2024-02-06, 16:43 IST

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

2 points

Arrays A and B each contain N integers arranged in a random sequence. We want to check if A and B have any entries in common. Which of the following would be the most efficient algorithm, asymptotically?

- For each pair of positions $0 \leq i,j < N$, check if A[i] is equal to B[j].
- Sort A using merge sort. For each element B[i], try to insert B[i] in A and see if you encounter a duplicate.
- Sort A using quicksort. For each element B[i], use binary search to check if B[i] appears in A.
- (•) Sort A and B using merge sort. Merge A and B to check for duplicates.

2 points

Suppose our aim is to sort an array in ascending order. Which of the following statements is true?

- (•) Input in descending order is worst case for both selection sort and insertion sort.
- Input in descending order is worst case for selection sort but not for insertion sort.
- Input in ascending order is worst case for both selection sort and insertion sort.
- Input in ascending order is worst case for insertion sort but not for selection sort.

2 points

Suppose we want to sort an array in descending order and we implement quicksort so that we always choose the last element in the array as the pivot element. Assume that the input is a permutation of {1, 2, …, n}. Which of the following would ***definitely*** be a worst case permutation of this input for this implementation of quicksort?

- {1, 2, …, n} with all odd numbers in ascending order followed by all even numbers in random order
- {1, 2, …, n} with all even numbers in descending order followed by all odd numbers in ascending order.
- (•) {1, 2, …, n} in descending order.
- {1, 2, …, n} in some random order.

2 points

Which of the following statements is ***not*** true?

- Quicksort and merge sort are both examples of divide and conquer algorithms.
- If we randomly choose a pivot element each time, quicksort will always terminate in time O(n log n).
- For every fixed strategy to choose a pivot for quicksort, we can construct a worst case input that requires time $O(n^2)$.
- If we could find the median in time O(n), quicksort would have worst case complexity O(n log n).

2 points

We have a list of pairs [("Ashwin",69),("Sumati",87),("Tanuja",69),("Brinda",87), ("Shabana",72),("Vijay",60)], where each pair consists of a student's name and his/her marks in a course. We sort these pairs in ascending order of marks. Which of the folllowing corresponds to a stable sort of this input?

- [("Vijay",60),("Tanuja",69),("Ashwin",69),("Shabana",72).("Sumati",87),("Brinda",87)]
- [("Vijay",60),("Ashwin",69),("Tanuja",69),("Shabana",72),("Sumati",87),("Brinda",87)]
- [("Vijay",60),("Tanuja",69),("Ashwin",69),("Shabana",72),("Brinda",87),("Sumati",87)]
- [("Vijay",60),("Ashwin",69),("Tanuja",69),("Shabana",72),("Brinda",87),("Sumati",87)]