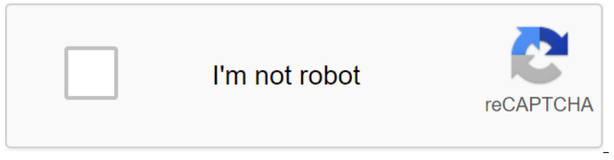

Continue

Sorting algorithms mcq pdf

Catalogue of objective questions covering all topics of computer science. Here you can access and discuss a few selection questions and answers to various compilation exams and interviews. What is a relapse for the worst case of quickSort and what is the complexity of time at worst? Solution: The worst case of quickSort occurs when the chosen rod is always one of the corner elements in the sorted array. At worst, quickSort re-calls one subproblem with a size 0 and another subproblem with size (n-1). Thus, the repetition of T (n) T (n-1) - T (0) - O (n) The aforementioned expression can be rewritten as T (n) T (n-1) temperature for; For zb; The zb and the tempo; - int partition (int arr', int si, int ei) int i q (si - 1); int j; for (j s si; j'ei - 1; j') exchange (Sarray, zarrej j); Exchange (Sarriai and 1), sarrei); Return (i No 1); - Implementation of fast sort arr -- Array, which will be sorted si -- Start index ei -- End of index (anta th e-arr, int si, int ei) - int pi; quickSort (arr, si, pi - 1); quickSort (arr, pi No 1, hey); This section focuses on sorting the data structure. These multiple choice (mcq) questions should be practiced to improve the data structure skills required for various interviews (campus interviews, walk-in interviews, company interviews), placement, entrance exam and other competitive exams. 1. What is repetition for the worst case of quickSort and what is the complexity of time at worst? A. Repetition of T (n) - T (n-2) - O (n), and time complexity - O (n'2) B. Repetition T (n) - T (n-1) - O (n) and time complexity - O (n'2) C. Repetition T (n) No 2T (n/2) - O (n) and Time Complexity O (nLogn) D. Repetition T (n) - T (n/10) - T (9n/10) - O (n) and Time Complexity O (nLogn) View Answers Ans : B Explanation: No Explanation. 2. Which of the following is not a stable sorting algorithm? A. Insert of the sort of B. Choice of the kind of C. Bubble sort of D. Merge of the kind of View Answers Ans : B Explanation:Choice of kind is not a stable sorting algorithm. 3. What is an external sorting algorithm? A. An algorithm that uses tape or disk during sorting B. An algorithm that uses basic memory while sorting C. Algorithm, which includes replacing the D. algorithm, which are considered in place view Answer Ans : Explanation: As the name suggests, the external sorting algorithm uses external memory as a tape or disk. 4. If the number of entries to be sorted is small, then..... sorting can be effective. A. Merger B. Heap C. Choice D. Bubble View Answer Ans : C Explanation: Sorting Choices Can Be Effective. 5.Suppose we have an O(n time algorithm) that finds unsorted array. Now let's look at the implementation of quickSort, where we first find the median using the above algorithm, and then use the median as a rod. What would be the worst case scenario being the complexity of this modified quickSort. A. O (n'2 Logn) B. O (n'2) C. O (n Logn Logn) D. O (nLogn) Ans Answers View : D Explanation: If we use the median as a rod element, then repetition for all cases becomes T (n) 2T (n/2) It falls in the case of 2 master method. 6. Which of the following is not an on-site sorting algorithm? A. Choosing a kind of B. Heap of the sort C. Fast Grade D. Merger of the sort View Answers Ans : D Explanation: Sorting merging is not a sorting algorithm in place. 7. What is the advantage of sorting bubbles over other sorting methods? A. Faster B. Consumes less memory C. Detects whether the input has already been sorted D. All mentioned Views Answer Ans : C Explanation: Bubble kind is one of the simplest sorting methods and perhaps the only advantage it has compared to other methods is that it can detect whether the input is already sorted. 8.Complexity of the sorting algorithm measures ..... as a function of the number n of items that should be sorted. A. Average time B. Time time time C. Average case complexity D. Case-complexity View Answer Ans : B Explanation: Complexity of the sorting algorithm measures the time of operation as a function of the number n of items that should be sorters. 9. Suppose we sort an array of eight whole rows using quicksort, and we just finished the first section with an array of looks like this: 2 5 1 7 12 11 10 What statement is correct? A. The rod can be either 7 or 9. B. The turn may be 7, but it's not 9 C. Turn is not 7, but it can be 9 D. Neither 7 nor 9 is the linchpin. View Ans responses: Explanation:7 and 9 are both in the right position (as in the sorted array). In addition, all the items left of 7 and 9 are smaller than 7 and 9 respectively and on the right are larger than 7 and 9 respectively. 10.Consider a situation in which the appointment operation is very expensive. Which of the following sorting algorithms should be performed in such a way that the number of destination operations is minimized as a whole? A. Insert of the Sort B. Choice of kind C. A bunch of sort D. No Viewing Answers Ans : B Explanation: Choice of kind. Also Check: Discussion Here's an array of ten wholes: 5 3 8 9 1 7 0 2 6 4 Draw this array after the first iteration of the large cycle in the sorting of choice (sorting from the smallest to the largest). Here's an array of ten wholes: 5 3 8 9 1 7 0 2 6 4 Draw this array after the first iteration of the large cycle in the insert sorting (sort from the smallest to the largest). This iteration has shifted at least one element into the array! that you write a program that has this choice of static method available: void selectionsort (int) (int) Data, first, int n); Your program also has an array called x, with 10 elements. Write two activation methods: First activation uses selectionsort to sort all x; The second call uses selectionsort to sort x.3. x.9. Short Answers Section 12.2 Recursive Sorting Algorithms Describe a Case where quicksort will result in square behavior. Here is an array that has just been divided in the first stage of the quicksort: 3, 0, 2, 4, 5, 8, 7, 6, 9 Which of these elements can be the rod? (There may be more than one opportunity!) Give a brief, accurate description of a good way for quicksort to choose a turning item. Your approach should be better than using a recording in a location. Give a brief, accurate description of a good way for quicksort to improve your performance with insertionsort. Here's an array of ten wholes: 5 3 8 9 1 7 0 2 6 4 Suppose we split this array using the quicksort section function and using 5 to rotate. Draw the resulting array after the section is complete. Here's an array of ten wholes: 5 3 8 9 1 7 0 2 6 4 Draw this array after completing two recursive merger sorting calls and before the final merger step. Implementation of the next static method: private static fusion of voids (int, int first, int n1, int n2); Premise: The data has at least n1'n2 components, starting with the data. The first n1 elements (from data to / data)first'n1-1) are sorted from the smallest to the largest, and the last elements (from the data firstn1 to the data firstn1'n2-1)) are also / sorted from the smallest to the largest. Postcondition: Starting with the data, the n1'n2 data elements have been rebuilt to be sorted from the smallest to the largest. Brief Replies Section 12.3 Algorithm O (n log n) Using a Heap Write two or three clear sentences to describe how a heap works. Fill in the next table on time to sort the array of n items. Use only big-o notation, and don't have any extraneous constants in expressions. Worst case is the average case of binary search sorted array .. Sorting inserts .. Sorting merger .. A quick variety without the median of the three rod choices.. A quick variety with an average of three rod selections.. Sorting of choice .. A bunch of kind.. (This question may not have been covered in your classroom.) Suppose you don't implement quicksort with a stack like in the last programming job. You use your algorithm to sort an array of 100 items, and at the beginning of the final iteration of the cycle while the stack contains only two numbers: 10 (top) and 90 (bottom). Write one or two clear sentences to describe which parts of the array are being sorted at the moment and which parts of the array have not yet been sorted. There are 58 questions to complete. This set The Structure Multiple Choice and Answers (MC) focuses on Bubble Sort. 1. What is it external sorting algorithm? (a) An algorithm that uses tape or drive during sorting (b) An algorithm that uses basic memory while sorting (c) An algorithm that includes a replacement (d) Algorithm that is considered in place View AnswerAnswer: Explanation: As the name suggests, the external sorting algorithm uses external memory as a tape or disk. 2. What is an internal sorting algorithm? (a) An algorithm that uses tape or drive during sorting (b) An algorithm that uses basic memory during sorting (c) An algorithm that includes a replacement (d) Algorithm that is considered in place view AnswerAnswer: b Explanation: As the name suggests, the internal sorting algorithm uses internal core memory. 3. What is the worst complexity of the bubble of sorts? (a) O (nlogn) b) O (logn) c) O (n) d) O (n2) View AnswerAnswer: d Explanation: Sorting the bubble works starting with the first element and replacing items if necessary in each iteration. 4. Choose the appropriate code that sortes the bubbles. (a) for (int j'arr.length-1; j'0; j---) for (int k'0; k'lt;j; k)) - int temp - arr'k; arr'k ' arr'k'1'; arr'k'1' - temp; - b) for (int j'arr.length-1; j'0; j'---) for (int k'0; zlt; z) if(arr)k't; arr'th; arr'k'1; arr'k'1; arr'k'rr'; arr'k'1; arr'k'1; arr'k'rr'; arr'k'1; arr'k'1; arr'k'rr'; arr'k'1; arr'k'rr'; arr'k'1; arr'k'rr'; arr'k'1; arr'k'rr'; arr'k'1; arr'k'rr'; arr'k'1; arr'k'rr'1; arr'k'rr'; arr'k'1; arr'k'arr'; arr'k'k'1; arr'k'1; arr'k'rr'k'1; arr'k'1'k'rr'; arr'k'1; arr'k'rr'; arr'k'rr'; arr'k'1; arr'k'1; arr'k'1; arr'k'rr'; arr'k'1' = }= }= c)= for(int= j=arr.length; j=> k<j; k++)= {= if(arr[k]=> arr'k'1)) - int temp - arr'k' ; arr'k ' arr'k'1'; arr'k '1' - temp; - d) for (int j'arr.length; j'1)0; j--) for (int k'0; k'lt;j; k)) q if (arr'kt; arr'k'2)) View AnswerAnswer: Explanation: The External Cycle tracks the number of iterations, and the internal cycle checks whether a replacement is needed. 5. What is the average complexity of a case of bubble kind? a) O (nlogn) b) O (logn) c) O (n) d) O (n2) View AnswerAnswer: d Explanation: Sorting the bubble works starting with the first element and replacing items if required in each iteration, even in the middle case. 6. Which of the following is not the advantage of an optimized bubble to sort over other sorting methods in the event of sorted items? a) This faster b) Consumes less memory in) Detects whether the input has already been sorted d) Consumes less time View AnswerAnswer: c Explanation: Optimized bubble view is one of the simplest sorting methods and perhaps the only advantage it has compared to other methods is that it can detect whether the input is already sorted. This is faster than others in the case of a sorted array and takes less time to describe whether the input array is sorted or not. It consumes the same memory as other sorting methods. Therefore, this is not an advantage. 7. This array is arr No. 1, 2, 4, 3 . Sorting bubbles is used for elements of the array. How many iterations will be done to sort the array? (a) 4 b) 2 p) 1 g) 0 Responses View: Explanation: Even if the first two elements have already been sorted, sorting requires 4 iterations to sort this array. 8. How can you improve the effectiveness of the best case in a bubble of sorts? (Entry has already been sorted) (a) boolean has been replaced by false; for (int j'arr.length-1; j'0' swapped; j---) - exchanges the truth; for (int k'0; k'lt;j; k)) for (int j'arr.length-1; j'0' swapped; j---) - replaced - false; for (int k'0; k'lt;j; if (int j'arr'k'1) - int temp - arr'k; arr'k 'arr'k'1;arr'k for (int j'arr.length-1; j'0' swapped; j---) - replaced - false; for (int k'0; k'lt; k'lt; if (arr'k'1) - int temp - arr'k; arr'k 'arr'k'; arr'k '1'; arr'k'1) - temp; for (int j'arr.length-1; j'0 'exchange; j---) for (int k'0; k'lt;j; k)) if (arr'k); arr'k'1) View AnswerAnswer: c Explanation: c Bullish variable 'swapped' determines whether any exchange has occurred in a particular iteration, if there has been no replacement, this array is sorted and no more iterations are required. 9. What is the best case of bladder efficiency of sorts in an impromptu version? (a) O (nlogn) b) O (logn) c) O (n) d) O (n2) View AnswerAnswer: c Explanation: Some iterations may be missed if the list is sorted, hence efficiency increases to O(n). 10. This array is arr (1,2,4,3). Sorting bubbles is used to sort array items. How many iterations will be done to sort the array with the improvised version? a) 4 b) 2 (c) 1 d) 0 View AnswerAnswer: b Explanation: Only 2 elements in this array are not sorted, hence only 2 iterations are required for sorting them. Sanfoundry Global Education - Learning Series - Data Structures and Algorithms. To practice all areas of data structures and algorithms, here's a complete set of 1000 Multiple question-and-answer choices. Participate in the Sanfoundry Certification Competition to receive a free certificate of merit. Join our social networks below and stay up to date with the latest contests, videos, internships and vacancies! Manish Bhojasia, a technology veteran from 20 years and Cisco Wipro, is the founder and CTO at Sanfoundry. He is a Linux Kernel Developer and SAN Architect and is passionate about developing competencies in these areas. He lives in Bangalore and conducts focused training for IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals and Storage Protocols such as iSCSI and Fiber Channel. Stay in touch with him - LinkedIn LinkedIn sorting algorithms mcqs. sorting algorithms mcqs pdf. sorting algorithms in data structures mcq. which of the following sorting algorithms is the fastest mcq. sorting algorithms in c mcq