QB365 Question Bank Software Study Materials

Algorithmic Strategies Important 2, 3 & 5 Marks Questions With Answers (Book Back and Creative)

12th Standard

Computer Science

Total Marks : 50

2 Marks

10 x 2 = 20

1) What is an Algorithm?

Answer : An algorithm is a finite set of instructions to accomplish a particular task. It is a step-by-step procedure for solving a given problem.

2) What is Sorting?

Answer : Sorting is a method of arranging group of items in ascending or descending order. Various sorting techniques in algorithms are Bubble sort, Quick sort, Heap sort, Selection sort, Insertion sort.

3) What is searching? Write its types.

Answer : A searching algorithm is the step-by step procedure used to locate specific data among a collection of data. There are two types of searching are.
(i) Linear Search
(ii) Binary Search

4) Write the phase of performance evaluation of an algorithm.

Answer : Analysis of algorithms and performance evaluation can be divided into two different phases:
(i) A Priori estimates: This is a theoretical performance analysis of an algorithm. Efficiency of an algorithm is measured by assuming the external factors.
(ii) A Posterori testing: This is called performance measurement. In this analysis, actual statistics like running time and required for the algorithm executions are collected.

5) What is Insertion Sort?

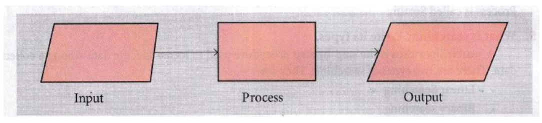
Answer : (i) Insertion sort is a simple sorting algorithm.
(ii) It works by taking elements from the list one by one and inserting them in their correct position into a new sorted list.
(iii) This algorithm builds the final sorted array at the end.

6) What is Definiteness?

Answer : All operations should be well defined for example operations involving division by zero or taking Square root for negative number are unacceptable.

7) Draw a Pictoral representation of typical algorithm.

Answer :



8) Define Big O.

Answer : Big O is often used to describe the worst-case of an algorithm.

9) What is selection Sort algorithm?

Answer : The selection sort is a simple sorting algorithm that improves on the performance of bubble sort by making only one exchange for every pass through the list. This algorithm will first find the smallest elements in array and swap it with the element in the first position of an array, then it will the second smallest element and swap that element with the element in the second position, and it will continue until the entire array is sorted in respective order.

10) Define memorization.

Answer : Memorization or memorisation is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls and returning the cached result when the same inputs occur again.

3 Marks

10 x 3 = 30

11) List the characteristics of an algorithm.

Answer : (i) Input
(ii) Output
(iii) Finiteness
(iv) Definiteness
(v) Effectiveness
(vi) Correctness
(vii) Simplicity
(viii) Unambiguous
(ix) Feasibility
(x) Portable
(xi) Independent

12) Discuss about Algorithmic complexity and its types.

Answer : The complexity of an algorithm f (n) gives the running time and/or the storage space required by the algorithm in terms of n as the size of input data.
(i) Time Complexity: The Time complexity of an algorithm is given by the number of steps taken by the algorithm to complete the process.
(ii) Space Complexity: Space complexity of an algorithm is the amount of memory required to run to its completion.

13) What are the factors that influence time and space complexity.

Answer : (i) **Time Factor** -Time is measured by counting the number of key operations like comparisons in the sorting algorithm.
(ii) **Space Factor** -Space is measured by the maximum memory space required by the algorithm.

14) Write a note on Asymptotic notation.

Answer : Asymptotic Notations are languages that uses meaningful statements about time and space complexity. The following three asymptotic notations are mostly used to represent time complexity of algorithms:
(i) Big O: Big O is often used to describe the worst -case of an algorithm.
(ii) Big Ω: Big Omega is the reverse Big O, if Big O is used to describe the upper bound (worst - case) of a asymptotic function, Big Omega is used to describe the lower bound (best -case).
(iii) Big Θ: When an algorithm has complexity with lower bound = upper bound, Say that an algorithm has a complexity O(n log n) and Ω (n log n), it's actually has the complexity Θ (n log n), which means the running time of that algorithm always falls in n log n in the best-case and worst-case.

15) What do you understand by Dynamic programming?

Answer : (i) Dynamic programming is an algorithmic design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions.
(ii) Dynamic programming approach is similar to divide and conquer. The given problem is divided into smaller and yet smaller possible sub-problems.
(iii) Dynamic programming is used whenever problems can be divided into similar sub-problems. So that their results can be re-used to complete the process.
(iv) Dynamic programming approaches are used to find the solution in optimized way. For every inner subproblem, dynamic algorithm will try to check the results of the previously solved sub-problems. The solutions of overlapped sub-problems are combined in order to get the better solution.

16) Write the different factors in which the time efficiency of an algorithm its measured

Answer : The execution time that you measure in this case would depend on a number of factors such as:

(i) Speed of the machine

(ii) Compiler and other system Software tools

(iii) Operating System

(iv) Programming language used

(v) Volume of data required

17) Write a pseudo code for Insertion sort.

Answer : **Step 1** - If it is the first element, it is already sorted.

Step 2 - Pick next element

Step 3 - Compare with all elements in the sorted sub-list

Step 4 - Shift all the elements in the sorted sub-list that is greater than the value to be sorted

Step 5 - Insert the value

Step 6 - Repeat until list is sorted

18) what are the phases available in analysis of algorithm?

Answer : Analysis of algorithms and performance evaluation can be divided into two different phases:

1) A Priori estimates:

This is a theoretical performance analysis of an algorithm. Efficiency of an algorithm is measured by assuming the external factors.

2) A Posteriori testing:

This is called performance measurement. In this analysis, actual statistics like running time and required for the algorithm executions are collected.

19) Write an algorithm to find the mean.

Answer : **SOURCE CODE:**

1. Enter the given numbers.
2. Set a total to 0.
3. Set a counter to 0.
4. Start with the first number in the list

a. Add the number to the total

b. Add one to the counter

c. Is this the last number in the list?

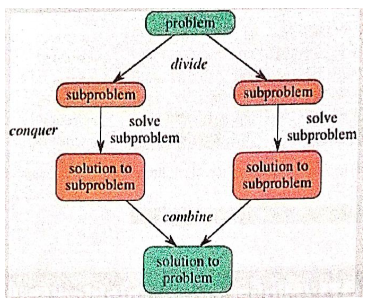
i. If no go to the next number and repeat steps a to c

ii. If yes go to step 5

5. Find the mean by dividing the total by the value of the counter.
6. Display mean

20) Divide and conquer algorithm.

Answer : **BINARY SEARCH:**



<u>5 Marks</u> 5 x 5 = 25

21) Explain the characteristics of an algorithm.

Answer :

Input	Zero or more quantities to be supplied.
Output	At least one quantityis produced.
Finiteness	Algorithms must terminate after finite number of steps.
Definiteness	All operations should be well defined. For example operations involving division by zero or taking square root for negative number are unacceptable.
Effectiveness	Every instruction must be carried out effectively.
Correctness	The algorithms should be error free.
Simplicity	East to implement.
Unambiguous	Algorithm should be clear and unambiguous. Each of its steps and their inputs/outputs should be clear and must lead to only one meaning.
Feasibility	Should be feasible with the avaliable resources.
Portable	An algorithm should be generic, independent of any programming language or an operating system able to handle all range of inputs.
Independent	An algorithm should have step-by-step directions, which should be independent of any programming code.

22) Discuss about Linear search algorithm.

Answer : (i) Linear search also called sequential search is a sequential method for finding a particular value in a list.

(ii) This method checks the search element with each element in sequence until the desired element is found or the list is exhausted. In this searching algorithm, list need not be ordered.

procedure :

1. Traverse the array using for loop
2. In every iteration, compare the target search key value with the current value of the list.

(i) If the values do not match, move on to the next array element.

(ii) If the values match, display the current index and value of the array.

3. If no match is found, display the search element not found.

(iii) To search the number 25 in the array given below, a linear search will go step by step in a sequential order starting from the first element in the given array if the search element is found that index is returned otherwise the search is continued till the last index of the array. In this example number 25 is found at index number 3.

index	0	1	2	3	4
values	10	12	20	25	30

Example 1:

Input: values[] = {5, 34, 65, 12, 77, 35)

target = 77

Output: 4

Example 2:

Input: values[] = [101, 392, 1, 54, 32, 22, 90, 93) target = 200

Output: -1 (not found)

23) What is Binary search? Discuss with example

Answer : Binary Search:

Binary search also called half-interval search algorithm. It finds the position of a search element within a sorted array. The binary search algorithm can be done as a divide- and -conquer search algorithm and executes in logarithmic time.

Pseudo Code:

Start with the middle element:

(i) If the search element is equal to the middle element of the array i.e., the middle value = number of elements in array/2, then return the index of the middle element.

(ii) If not, then compare the middle element with the search value,

(iii) If the search element is greater than the number in the middle index, then select the elements to the right side of the middle index, and go to Step-1.

(iv) If the search element is less than the number in the middle index, then select the elements to the left side of the middle index, and start with Step-1.

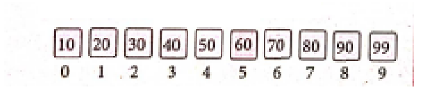
(v) When a match is found, display success message with the index of the element matched.

(vi) If no match is found for all comparisons, then display unsuccessful message.

Binary Search Working principles :

(i) List of elements in an array must be sorted first for Binary search. The following example describes the step by step operation of binary search.

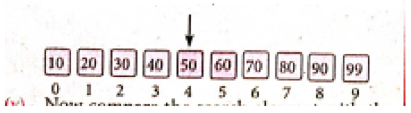
(ii) Consider the following array of elements, the array. is being sorted so itenables to do the binary searçh algorithm. Let us assume that the search element is 60 and we need to search the location or index of search element 60 using binary search.



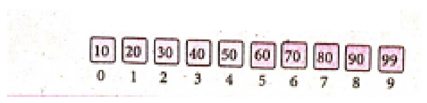
(iii) First, we find index of middle element of. the array byusing this formula:

mid = low + (high - low) /2

(iv) Here it is, 0 + (9-0)/2=4 (fractional part ignored). So, 4 is thè mid value of the array.



(v) Now compare the search element with the value stored at mid value location 4. The value stored at location or index 4 is 50, which is not match with search element. As the search value 60 is greater than 50.

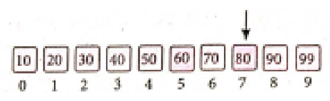


(vi) Now we change our low to mid+1 and find the new mid value again using the formula.

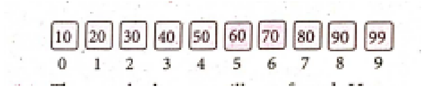
low = mid + 1

mid = low + (high - low) / 2

(vii) Our new mid is 7 now. We compare the value stored at location 7 with our target value 60.



(viii) The value stored at location or index 7 is not a match with search element, rather it is more than what we are looking for. So, the search element must be in the lower part from the current mid value location.

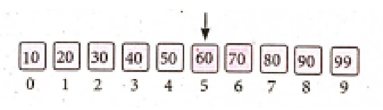


(ix) The search element still not found. Hence, we calculated the mid again by using the formula.

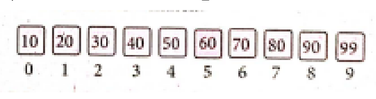
high = mid -1

mid = low +(high - low)/2

Now the mid value is 5.



(x) Now we compare the value stored at location 5 with our search element. We found that it is a match.



(xi) We can conclude that the search element 60 is found at locationor index 5. For example if we take the search element as 95, For this value this binary search algorithm return unsucessful result.

24) Explain the Bubble sort algorithm with example.

Answer : Bubble sort algorithm:

(i) Bubble sort algorithm simple sorting algorithm. The algorithm starts at the beginning of the list of values stored in an array. It compares each pair of adjacent elements and swaps them if they are in the unsorted order.

(ii) This comparison and passed to be continued until no swaps are needed, which indicates that the list of values stored in an array is sorted. The algorithm is a comparison sort, is named for the way smaller elements "bubble" to the top of the list.

(iii) Although the algorithm is simple, it is too slow and less efficient when compared to insertion sort and other sorting methods.

(iv) Assume list is an array of n elements. The swap function swaps the values of the given array elements.

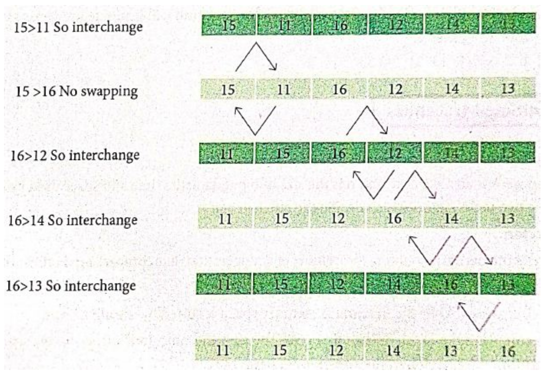
Procedure :

(i) Start with the first element i.e., index = 0, compare the current element with the next element of the array.

(ii) If the current element is greater than the next element of the array, swap them.

(iii) If the current element is less than the next or right side of the element, move to the next element. Go to Step 1 and repeat until the end of the index is reached.

(iv) Let's consider an array with values {15, 11, 16, 12, 14, 13} Below, we have a pictorial representation of how bubble sort will sort the given array.



(v) The above pictorial example is for iteration-d. Similarly, remaining iteration can be done. The final iteration will give the sorted array. At the end of all the iterations we will get the sorted values in an array as given below:

11	12	13	14	15	16

25) Explain the concept of Dynamic programming with suitable example.

Answer : (i) Dynamic programming is an algorithmic design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions.

(ii) Dynamic programming approach is similar to divide and conquer. The given problem is divided into smaller and yet smaller possible sub-problems.

(iii) Dynamic programming is used whenever problems can be divided into similar subproblems. so that their results can be reused to complete the process.

(iv) Dynamic programming approaches are used to find the solution in optimized way. For every inner subproblem, dynamic algorithm will try to check the results of the previously solved sub-problems.

(v) The solutions of overlapped sub-problems are combined in order to get a better solution.

Steps to doDynamic programming :

(i) The given problem will be divided into smaller overlapping sub-problems.

(ii) An optimum solution for the given problem can be achieved by using result of smaller sub-problem.

(iii) Dynamic algorithms uses Memoization

Fibonacci Series - An example :

(i) Fibonacci series generates the subsequent number by adding two previous numbers. Fibonacci series starts from two numbers -Fib 0 & Fib 1. The initial values of Fib 0 & Fib l can be taken as 0 and 1.

(ii) Fibonacci series satisfies the following conditions:

Fibn = Fib_{a-1} + Fib_{a-2}

(iii) Hence, a Fibonacci series for the n value 8 can look like this

Fib_8 = 0 1 1 2 3 5 8 13

Fibonaeci Iterative Algorithm with Dynamic programning approach : The following example shows a simple Dynamic programning approach for the generation ot Fibonacci series.

Initialize f0 = 0, f1 = 1.

Step- 1: Print the initial values of Fibonacci f0 and f1

Step- 2: Calculate Fibonacci fib $\leftarrow$ f0+ f1

Step- 3: Assign f0 $\leftarrow$ f1, f1 $\leftarrow$ fib

Step- 4: Print the next consecutive value of Fibonacci fib

Step- 5: Go to step-2 and repeat until the specified number of terms generated

Example:

if we generate Fibonacci series up to 10 digits, the algorithm will generate the series as shown below:

The Fibonacci series is:

0 1 1 2 3 5 8 13 21 34 55.