**CS502 SOLVED MCQs**

The sequence of merge sort algorithm is:

* Divide Combine-Conquer
* Conquer-Divide-Combine
* **c. Divide-Conquer-Combine             Page 27**
* Combine-Divide-Conquer

 In \_\_\_\_\_\_ Knapsack Problem, limitation is that an item can either be put in the bag or not. Fractional items are not allowed.

* 0
* 1
* **0/1                                                                                Page 91**
* Fractional

 In Selection algorithm, we assume pivot selection takes theta \_\_\_\_\_\_\_ running time.

* **a. n                                                                     Page – 36**
* n2
* n3
* log (n)

 In Heap Sort algorithm (using max heap), when every time maximum elements removed from top \_\_\_\_\_\_\_\_.

* We call merge Sort Algorithm
* it becomes Order n2 Algorithm
* Divide and Conquer strategy helps us
* **d. We are left with a hole                      Page – 41**

 If matrix A of dimension p x q is multiply with matrix B of dimension q x r, then each entry in resultant matrix takes \_\_\_\_\_\_\_ time.

* a**. O (q)                                                                          Page – 84**
* O (1)
* O (p x q)
* O (q x r)

 \_\_\_\_\_\_\_\_\_ is a method of solving a problem in which we check all possible solutions to the problem to find the solution we need.

* Plane-Sweep Algorithm
* Sorting Algorithm
* c**. Brute-Force Algorithm                      google**
* Greedy approach

 The worst case running time of Quick sort algorithm \_\_\_\_\_.

* Cannot be quadratic
* **b. Is quadratic**
* ls always Exponential
* Is linear

 In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with \_\_\_\_\_ leaf in the tree.

* second last
* **b. Last                                                                           Page -41**
* First
* Any

 Quick sort algorithm was developed by –

* AlferdAho
* Sedgewick
* John Vincent Atanasoff
* **d. Tony Hoare                                                     – Google wikipedia**

 If Matrix-A has dimensions “3×2” and Matrix-B has dimensions “2×3”, then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.

* 3×2
* 2×3
* 2×2
* **d. 3×3**

 For comparison-based sorting algorithms, it is possible to sort more efficiently than Omega n log(n) time.

* Always
* ·        Sometimes not
* **NOT           Pg 54**
* Sometimes

 Dynamic Programming approach is usually useful in solving optimization problems.

* **True**
* False

 In Sorting the key value or attribute\_\_\_\_\_ from an ordered domain.

* **Must be                        page 39**
* Not always
* May be
* Occasionally

 Result of asymptotical analysis of n(n -3) and 4n\*n is that \_\_\_\_\_\_\_

* n(n-1) is asymptotically Less
* n(n-1) is asymptotically Greater
* Both are asymptotically Not equivalent
* **Both are asymptotically Equivalent       page 23  (4n\*n= 4n2)**

 Floor and ceiling are \_\_\_\_\_\_ to calculate while analyzing algorithms

* Very easy
* 3rd Option is missing
  + **Usually considered difficult**
* 4th Option is missing

 \_\_\_\_\_ of reference is an important fact of current processor technology.

* Defining
* Assigning
  + **Locality   pg-8**
* Formality

 In max-heap, largest element is stored at root node. Where is the smallest element stored?

* **Right Node**
* Leaf Node           Not sure
* Middle Node
* Left Node

 In average-case time analysis of Quick sort algorithm, the most balanced case for partition is when we divide the list of elements into \_.

* **Equal no. of pieces as of input elements**
* Single piece exactly
* Two nearly equal pieces
* Three nearly equal pieces

 Which of the following is calculated with Big O notation?

* Medium bounds
* **Upper bounds                                    Page – 25**
* Lower bounds
* Both upper and lower bounds

 Edit distance algorithm based on \_\_\_\_\_\_\_\_ strategy

* Greedy
* **Dynamic Programming                         Page – 81**
* Divide and Conquer
* Searching

 In Heapsort Algorithm, total time taken by heapify procedure is \_\_\_\_\_\_\_\_

* **O (log n)                                                                      Page-43**
* O (log2 n)
* O (n log n)
* O (n2 log n)

 Al-Khwarizmi was a/an \_\_\_\_\_\_\_

* Artist
* Astronomer
  + **Mathematication p-7**
* Khalifah

 When matrix A of 5x3is multiply with metric B of 3×4 then the number of multiplication required is: Not found exactly

* 15
* 12
* 36
* **60     Not Found exactly but as per formula at page 84,**

 Pseudo code of algorithms are to be read by \_\_\_\_\_\_\_.

* **People**Page -12
* RAM
* Computer
* Compiler

 The sieve technique is a special case, where the number of sub-problems is Just

\_\_\_\_\_\_\_\_\_

* **1                                                                          P-34**
* 2
* 3
* 4

When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has \_\_\_\_\_\_\_\_ sub-problems.

* **Overlapping                          – Google Search**
* Over costing
* Optimized
* Three

Sieve technique is very important special case of Divide-and-Conquer strategy.

* False
  + **True**

In order to say anything meaningful about our algorithms, it will be important for us to settle on a \_\_\_\_\_\_.

* Java Program
* C++ Program
  + **Mathematically model of computation**
* Pseudo program

Merge sort is based on \_\_\_\_\_\_\_.

* Brute-force
* Plan-sweep
  + **Divide and Conquer**
* Axis-sweep

 What time does Merge Sort algorithm take in order to sort an array of ‘n’ numbers?

* (n)
* (log n)
* (n^2)
* **d. (n log n) Google Search 31. In Heap Sort**

 algorithm, the first step is to \_\_\_\_\_\_\_\_\_\_\_.

* **Call Build-Heap procedure Page – 46**
* Sort the array in descending order
* Call Heapify procedure
* Find the number of input elements

The definition of theta-notation relies on proving \_\_\_\_\_\_\_\_ asymptotic bound.

* One
* Lower
* Upper
* **Both lower & upper                    Page – 25**

In merge sort algorithm, to merge two lists of size n/2 to a list of size n, takes

\_\_\_\_\_\_\_ time.

* **Theta (n)                                                                     Page – 32**
* Theta log(n)
* Theta log2(n)
* Theta n log(n)

 We can make \_\_\_\_\_\_\_ recursive calls in Fibonacci Sequence.

* Infinite
* **Finite                       google**
* Only one
* Zero

 Following is NOT the application of Edit Distance problem.

* Speech recognition
* Spelling Correction
* **Ascending Sort                                                                 Page – 76**
* Computational Molecular Biology

In plane sweep approach, a vertical line is swept across the 2d-plane and structure is used for holding the maximal points lying to the left of the sweep line.

* Array
* Queue
* **Stack                                                                                              Page – 18**
* Tree

When a heapify procedure is applied to the root node to restore the heap, then at each level, the comparison performed takes time:

* It will take (log n).
* It can not be predicted
* **It will take O (1).                                                               Page – 43**
* Time will vary according to the nature of input data.

\_\_\_\_\_ time is the maximum running time over all legal inputs.

* **Worst-case                                                                             Page – 13**
* Average-case
* Best-case
* Good-case

Efficient algorithm requires less computational…

* Memory
* Running Time
* **Memory and Running Time                               Page – 9**
* Energy

For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from \_\_\_\_\_\_

* half of the input values
* **all possible random values               Page – 50**
* Pivot is input separately
* values greater than 5

Selection algorithm takes theta \_\_\_\_\_\_

* (n2)
* **(n)                                                                                                        Page – 37**
* log(n)
* n log(n)

Recurrence can be described in terms of a tree.

* **Yes                                                                                                    Page – 31**
* No

Time complexity of Dynamic Programming based algorithm for computing the minimum cost of Chain Matrix Multiplication is \_\_\_\_\_\_

* Log n
* n
* n^2 (n square)
* **d. n^3 (n cube)                                                   Page -90**

The Iteration method is used for \_\_\_\_\_\_

* **Solving Recurrence relations                                        Page 31**
* Merging elements in Merge sort
* Comparing sorting algorithms only
* Dividing elements in Merge sort

In 3-Dimensional space, a point P has \_\_\_\_\_\_ coordinate(s).

* (X, Y)
* (X, 0)
* (0, Y)
* **(X,Y, Z)**

Chain matrix multiplication problem can be solved through \_\_\_\_\_\_ strategy.

* **Dynamic programming**                                                           Page – 85
* Greedy
* Divide and conquer
* Sorting

Merge sort have running time….running time of Heap sort. Not found exactly

* Greater than
* **Less than                                                  Google**
* Equal to
* Different than

Median is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

* True
* False                                                              Page – 34

We do not need to mathematically prove that for comparison-based sorting algorithms always takes Omega nlog (n) time.

* **True                                  Google**
* False

The Omega-notation allows us to state only the asymptotic \_\_\_\_\_\_ bounds.

* Middle
* **Lower                                                 Page 25**
* Upper
* Both lower & upper

Both lower & upperSorting can be in \_\_\_\_\_\_\_\_

* Increasing order only
* Decreasing order only
* **Both Increasing and Decreasing order**
  + Random order

Radix sort performs sorting the numbers \_\_\_\_\_\_\_ digit (s) at a time.

* **One                                                                                   Page – 71**
* Two
* Three
* All

Quicksort is a/an \_\_\_\_\_\_ and \_\_\_\_\_\_\_\_ sorting algorithm.

* Not in place, not stable one
* **In place , not stable one                                        Page – 54**
* In place , stable one
* Not in place , stable one

Consider three matrices X,Y,Z of dimensions 1×2, 2×3,3×4 respectively. The number of multiplications of (XY) Z is:

* **18                        As per lecture slides**
* 32
* 24
* 30

In Fibonacci Sequence, unnecessary repetitions do not exist at all.

* True
* **False                                                             Page – 74**

 It is not a Fibonacci sequence .                     1,1,1,2,3,5,8,13,21,34,55,…..

* **True                                                                                Page – 73**
* False

Heap sort is a/ an \_\_\_\_\_\_ and \_\_\_\_\_\_\_\_ sorting algorithem.

* Not in place, not stable one
* **In place , not stable one                                       Page – 54**
* In place , stable one
* Not in place , stable one

Identify the True Statement

* The knapsack problem does not belong to the domain of optimization problems.
* **The knapsack problem belongs to the domain of optimization**

**problems.                                                 Page – 91**

* The Knapsack problem cannot be solved by using dynamic programming
* The knapsack problem is optimally solved by using brute force algorithm.

In Dynamic Programming, our approach is to \_\_\_\_\_\_\_\_\_

* Develop the solution in a top-down fashion
* Express the problem non-recursively
* **Build the solution in a bottom-up fashion     Page – 75**
* Input several sub-problems simultaneously

Counting sort is suitable to sort the elements in range 1 to K;

* K is large
* **K is small                                                                  Page – 57**
* K may be large or small
* None

We can multiply two matrices A and B only when they are compatible which means

* **Number of columns in A must be equal to number of rows in B.**

**it seems Correct as per page 84**

* Number of columns in A must be equal to number of columns in B
* Number of rows and columns do not matter
* Number of rows in A must be equal to number of rows in B

Matrix multiplication is a (n) \_\_\_\_\_\_\_\_ operation.

* Commutative
* **Associative                                                            Page 85**
* Neither commutative nor associative
* Commutative but not associative

In Dynamic Programming approach, solution is modified / changed

* Always once
* **At each stage                                      google and wikipedia**
* Only for specific problems
* At 4th stage only

In Knapsack problem, the goal is to put items in the Knapsack such that the value of the items is \_\_\_\_\_\_\_\_\_\_ subject to weight limit of knapsack.

* Minimized
* Decreased
* **Maximized                                                               Page – 91**
* None of the given options

An in-place sorting algorithm is one that \_\_\_\_\_\_\_\_ uses additional array for storage.

* Always
* Permanently
* **Does not                                                                   Page – 54**
* Sometime

Memoization is a part of Dynamic Programming Strategy.

* **True                                                                                  Page – 74**
* False

If matrix A of dimension 2×4 is multiply with matrix B of dimension 4×3, then

the dimension of resultant matrix is      Not found exactly

* 2×4
* 4×3
* 3×4
* **2×3                     It seems correct as per second last Para of page 84**

In Dynamic Programming approach, we do not store the solution to each sub-problem in case if it reappears.

* True
* **False                                                                                                               Page – 75**

Dynamic Programming is a problem-solving approach in which\_\_\_

* Problem is solved in Zero time
* Solution is developed only at final stage
* Both are correct
* **Both are incorrect                                  google**

In Fibonacci sequence, each term is calculated by\_\_\_\_ previous\_\_ terms.

* Subtracting, Two
* Adding, Three
* **Adding, Two                                          Page – 73**
* Multiplying, Two

Selection sort is not an in-place sorting algorithm.

* **True                                                                                  Page – 54**
* False

If there are θ (n2) entries in edit distance matrix then the total running time is:

* θ (n)
* θ (1)
* **θ (n2)                                                                              Page – 84**
  + θ (n logn)

The only way to convert a string of i characters into the empty string is with i deletions, represented as

* E(0.j) =j
* E(i.j) = 1
* E(0.i) = j
* **E (i.0)=I                                                         Page – 78**

Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in an

* Array
* Variable
* **Table                                                                                                               Page – 86**
* class

We can use the optimal substructure property to devise a formulation of the edit distance problem.

* Selective
* Optimum
* Iterative
* **Recursive                                                                                 Page – 78**

Sorting is performed on the basis of \_\_\_\_\_\_\_\_\_\_\_.

* Computational resources
* Asymptotic notation
* Summation
* **Some key value of attribute                              page- 39**

 In Heap Sort algorithm, we call Build-heap procedure \_\_\_\_\_\_\_\_\_\_\_\_.

* **Only once                                                                 page 46**
* Twice
* Thrice
* As many times as we need

Radix sort is not a non-comparative integer sorting algorithm.

* **True                                                  Google Search**
* False

In the statement “output P[1].x, P[1].y”, the number of times elements of P are accessed is \_\_\_\_\_\_\_.

* 1
* **b. 2                                     page 14**
* 3
* 4

The main purpose of mathematical analysis is measuring the \_\_\_\_\_\_ required by the algorithm.

* Inputs & outputs
* Space
* Execution TIME page 13
* Execution time and memory

\_\_\_\_\_\_\_ provides us more accurate result when input values are not closer with each other

* Mode
  + Average
  + **Median  P-34**
* Mean

The process of \_\_\_\_\_\_ ends when you are left with such tiny pieces remaining that it is trivial to solve them.

* Brute-force
* Plan-sweep
* **Divide and Conquer**
* Axis-sweep

\_\_\_\_\_\_\_\_\_\_ overcomes the limitations of \_\_\_\_\_\_\_ by working as per positional notations of numbers.

* Counting sort, Radix sort
  + **Radix sort, Counting sort**

 Memorization is a part of Dynamic Programming strategy.

* False
  + **True**

Rank of an element can be defined as \_\_\_\_\_\_\_\_\_\_\_.

* One minus the number of elements that are smaller
* Two plus the number of elements that are greater
* **One plus the number of elements that are smaller P-34**
* Two minus the number of elements that are smaller

If the time complexity of an algorithm is given by O (1), then its time complexity would be

* Polynomial
* Exponential
* **Constant                                    – Wikipedia**
* Average

Quick sort is a recursive algorithm.

* **True**
* False

The asymptotic growth of n(n+1)/2 is:

* **O(n2)   As the n^2 term has the largest contribution, the Big-O complexity is O(n^2)**
  + O(n)
* O(n+2)
* O(n log n)

Approach of solving geometric problems by sweeping a line across the plane is called \_\_\_\_\_ sweep.

* Line
* **Plane              Page 18**
* Cube
* Box

As per algorithm of Dynamic Programing, we need to store

* First sub-problem only
* Best solution only
* **Intermediate sub-problems                               Pg:75**
* Final solution only

In Sieve technique, we solve the problem

* **In recursive manner                    Pg:34**
* Non recursively
* Using Merge Sort algorithm
* Using Brute force technique

One of the limitation in 0/1 knapsack is that an item can either be \_\_\_\_\_\_\_\_ in the bag or not.

* Use
* **Put                                                                                     Pg:91**
* Move
* Store

Which one is not passed as parameter in Quick sort algorithm?

* End of the array
* Middle of the array
* **Array (containing input elements)                           Google**
* Start of the array

In the analysis of Selection algorithm, we get the convergent \_\_\_\_\_\_\_\_\_

* Harmonic
* Linear
* Arithmetic
* **Geometric                                                                Pg:37**

A Random Access Machine (RAM)is an idealized machine withrandom access memory.

* **Infinite large                                          Pg:10**
* 512 MB
* 256 MB
* 2 GBs

While analyzing Selection algorithm, we make a number of passes, in fact it could be as many as

* n(n+1)
* **log(n)                                                                              Pg:37**
* n/3
* n/4

In Random Access Machine (RAM), instructions are executed in

* Parallel
* Batch
* **One by One                                                            Pg:10**
* Multiple times

In selection problem, the rank of an element will be its \_\_\_\_\_\_\_\_ position

* First
* **final                                                                                   Pg:34**
* Second last
* Last

The worst-case running time of Merge sort is \_\_\_\_\_ in order to sort an array of n elements.

* O(log n)
* O(n)
* **O(n log n)                                  page 40 and google**
* O(n)

f(n) and g(n) are asymptotically equivalent. This means that they have essentially the same \_\_\_\_\_\_.

* Results
* Variables
* **Growth Rates**
* Size

An algorithm is a mathematical entity. Which is independent of \_\_\_\_\_\_\_.

* Programming language
* Machine and Programming language
* **Programing Language Compiler and Machine**
* Compiler and Programming language

In Quick sort algorithm, Pivots form \_\_\_

* Stack
* Queue
* Graph
  + **Binary Search Tree**

Counting sort is suitable for sorting the elements within range 1 to P. where

* P is large
* P is very large
* P is undetermined
  + **P is Small**

In asymptotical analysis of n'(5 2)-3, as n becomes large, the dominant (fastest growing) term is some constant times

* n\_1
* n
* n+1
* **n\*n p-23**

\_\_\_ Items are not allowed in the 0/1  knapsack.

* Lighter
* Whole
* Weighty
* **Fractional**

Fibonacci Sequence was named on \_\_\_\_\_\_, a famous mathematician in 12th Century.

* Fred Brooks
* Grady Booch
* **Leonardo Pisano**
* Edgar F. Codd

In Heap Sort algorithm, we build \_\_\_\_\_ for ascending sort.

* Min heap
  + **Max Heap pg-41**

Bubble sort is not an in-place sorting algorithm.

1. True

|  |  |
| --- | --- |
| b. False | P-54 |

In partition algorithm, the subarray \_\_\_\_\_\_ has elements which are greater than pivot element x.

* A[p…r]
* A[p…q-1]
* A[q]
  + **S[q+1…r]**

In Heap Sort algorithm, if heap property is violated

* We call Heapify procedure
* We ignore
* Heap property can never be violated
* **We call Build Heap procedure**

\_\_\_\_\_\_ is not a characteristic of Random Access Machine.

* Assigning a value to a variable
* Locality of reference
* **Single-Processor      P-10**
* Executing an arithmetic instruction

The only way to convert an empty string into a sting of j characters is by doing j insertions, represented as \_\_\_\_\_\_

* E(i,j) = 1
* E(I,0) = I
* **E(0,j) = j                      page 78**
* E(1,j)= j

In Selection problem, the Sieve technique works in \_\_\_\_\_\_\_\_\_\_.

* Non-recursive manner
* Constant time
* **Phases         page 34**
* One complete go

Algorithm is a sequence of computational steps that —- the input into output.

* Merge
* Assign
* **Transform                page 7**
* Integrate

If pj dominates pi and pi dominates ph then pj also dominates ph, it means dominance relation is

* **Transitive                  page 18**
* Non Transitive
* Equation
* Symbolic

To find maximal points in brute-force algorithm each point of the space is compared against \_\_\_\_\_\_ of that space.

* One other point
* **All other points                       page 11**
* Few other points
* Most of the other points

In the following code the statement “cout<<j;”executes ——— times. for (j=1; j<=5; j = j+2)

cout<<j;

* 5 times
* 2 times
* **3 times**
* 0 times

In merge sort algorithm, we split the array around the \_\_\_\_\_\_ index q.

* **Mid                     page 17**
* Exiting
  + Entring
* Summing

In Selection problem, the Sieve technique \_\_\_\_\_\_\_\_\_.

* Add some more input items each time
* Do not work recursively
* Do not uses Divide and Conquer approach
* **Eliminates undesired data items each time**

Consider three matrices X, Y, Z of dimensions 1 x 2, 2 x 3, 3 x 4 respectively. The number of multiplications of X(YZ) is .

* 16
* 32
* 26
* **32                      page 84**

In Heap Sort algorithm, the total running time for Heapify procedure is

\_\_\_\_\_\_\_

* **Theta (log n)**
* Order (log n)
* Omega (log n)
* O(1) i.e. Constant time

The sieve technique works where we have to find\_\_\_\_\_\_\_ items(s) from a large input.

* **Single               page 34**
* Two
* Three
* Similar

In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object i , then we gain\_\_\_\_\_\_

* W(Total Weight of Knapsack)
* V (Total Value of all items)
* **vi (Value of object i)                      page 93**
* Nome of the given option

While Sorting, the order domain means for any two input elements x and y

\_\_\_\_\_\_\_ satisfies only.

* **x < y                             page 39**
* x > y
* x = y
* All of the above

For solving Selection problem, we introduced Sieve technique due to

\_\_\_\_\_\_\_

* **Using Decrease and Conquer strategy        page 34**
* Avoiding to sort all input data
* Eliminating Rank of an element
* Using Brute-force approach

\_\_\_\_\_\_\_\_ is one of the few problems, where provable lower bounds exist on how fast we can sort.

* Searching
* **Sorting                                page 38**
* Both Searching & sorting
* Growing

In plane sweep approach, a vertical line is swept across the 2d-plane from\_\_\_\_\_.

* Right to Left
* **Left to Right               page 18**
* Top to Bottom
* Bottom to top

In generating Fibonacci sequence, we can avoid unnecessary repetitions by

\_\_\_\_\_ process.

* Tokenization
* **Memorization               page 43**
* Randomization
* Memorization

For \_\_\_\_\_\_\_\_\_ values of n, any algorithm is fast enough.

* Medium
* Large
* **Small                                             page 14**
* Infinity

Dynamic programming comprises of \_\_\_\_\_\_\_.

* Recursion only
* Repetition only
* Recursion with Repetition
* **No Repetition but Recursion                 page 75**

The function f(n)=n(logn+1)/2 is asymptotically equalient t nlog n :Here Lower Bound means function f(n) grows asymptotically at \_\_ as fast as nlog n.

* Least                                   page 23
* Normal
* Most
* At

Counting sort has time complexity.

* O(n+k)
* **O(n)                                   page 58**
* O(k)
* O(nlogn)

Due to left complete nature of binary tree, the heap can be stored in

* **Array                                page 40**
* Structures
* Link List
* Stack

Single item from a larger set of \_\_\_\_\_\_\_\_.

* Constant
* Pointers
* Phases
* **n items                           page 34**

In the clique cover problem, for two vertices to be in the same group, they must be \_\_\_\_\_\_ each other.

* Apart from
* Far from
* Near to
* **Adjacent to                                 page 76**

How much time merge sort takes for an array of numbers?

* T(n^2)
* T(n)
* T(log n)
* **T(n log n)                                    page 40**

In in-place sorting algorithm is one that uses arrays for storage.

* **No additional array                                   page 54**
* An additional array
* Both of above may be true according to algorithm
* More than 3 arrays of one dimension

Brute-force algorithm for 2D-Maxima is operated by comparing \_\_\_\_\_\_

pairs of points.

* Two
* Some
* Most
* **All                   page 18**

While Sorting, the ordered domain means for any two input elements x and y \_\_\_\_ satisfies only.

* x > y
* x < y
* x = y
* **All of the above                    page 38**

Quick sort is.

* **Not stable but in place                          page 54**
* Stable but not in place
  + Stable & in Place
* Some time stable & some times in place

Which may be a stable sort?

* Merger
* Insertion
* **Both above                        page 54**
* None of the above

                For the Sieve Technique we take time.

* **T(nk)                         page 34**
* IT(n / 3)
* n^2
* n/3

Continuation sort is suitable to sort the elements in range 1 to k.

* K is Large
* K is not known
* K may be small or large
* **K is small                       page 54**

Asymptotic growth rate of the function is taken over \_\_\_\_\_\_ case running time. .

* **Worst                        page 14**
* Average
  + Best
* Normal

The sieve technique is a special case, where the number of sub problems is just.

* 5
* Many
* **1                 page 34**
* Few

In Quick sort, we don’t have the control over the sizes of recursive calls.

* **True                 page 49**
* False
* Less information to decide
* Ether true or false

Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their \_\_\_\_\_ coordinates. .

* Y
* Z
* **X**
* X , Y

Random access machine or RAM is a/an.

* Machine build by Al-Khwarizmi
* Mechanical machine
* **Mathematical model                    page 10**
* Electronics machine

The Huffman codes provide a method of encoding data inefficiently when coded using ASCII standard.

* True
* **False                                        page 99**

A heap is a left-complete binary tree that confirms to the \_\_\_\_\_\_\_\_.

* increasing order only
* decreasing order only
* **heap order                      page 40**
* log n order

If we associate (x, y) integers pair to cars where x is the speed of the car and y is the negation of the price. High y value for a car means a \_\_\_\_\_\_\_\_ car.

1. Fast

* Slow
* Expensive
* Cheap

Which one of the following sorting algorithms is the fastest?

* Merge sort
* **Quick sort**
* Insertion sort
* Heap sort

Quick sort algorithm divide the entire array into \_\_\_\_\_\_\_\_ sub arrays.

* **2**
* 3
* 4
* 5

In brute force algorithm, we measure running time T(n) based on \_\_\_\_\_\_\_\_.

* Worst-case time and best-case time
* **Worst-case time and average-case time                          page 46**
* Average-case time and best-case time
* Best-case time and staring-case time

For 2D Maxima problem. Plane Sweep algorithm first of all \_\_\_\_\_\_\_\_\_.

* **Sorts all points**
* Delete some points
* Output the elements
* Pushes all points on stack

There are \_\_\_\_\_\_\_\_ entries in the Edit Distance Matrix

* ϴ (n)
* **ϴ (n2)                        page 84**
* ϴ (n+2)
* ϴ (n + 100)

Which symbol is used for Omega notation?

* (O)
* (ϴ)
* **(Ω)**
* (@)

Selection sort is a \_\_\_\_\_\_ sorting algorithm

* **In-place                                 page 54**
* Not In-Place
* Stable
* in-partition

In Dynamic Programming based solution of knapsack problem, to compute entries of ‘V’, we will imply a(n) \_\_\_\_\_\_ approach.

* Subjective
* Inductive
* Brute Force
* Combination

We do not need to prove comparison-based sorting algorithms by mathematically. It always takes \_\_\_\_\_\_\_\_\_ time.

* Big Oh nlog(n)
* **Omega nlog(n)            NOT SURE**
* Omega n(n^2)
* Theta nlog(n)

Merge sort is a/an \_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_ sorting algorithm

* Not in-place, not stable one
* In-place, not stable one
* In-place, stable one
* **Not in-place, stable one                      page 54**

Cubic function will \_\_\_\_\_\_\_\_ a quadratic function.

* Prove
* be equal to
* **overtake                     Page 25**
* find

Insertion sort is a \_\_\_\_\_\_\_\_\_ sorting algorithm

* Unstable
* In-place       Page 54
* Not In-Place
* in-partition

To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is \_\_\_\_\_\_\_\_.

* Big-oh notation
* **Theta notation**
* Omega notation
* All of the given

Asymptotic growth of 8n^2 + 2n – 3 is:

* Θ(n^2 + n)
* **Θ (n^2)                      page 14**
* Θ(8n^2)
* Θ(8n^2 + 2n)

In the analysis of algorithms, \_\_\_\_\_\_\_\_\_ plays an important role.

* text analysis
* **time**
* growth rate
* money

In inductive approach of knapsack problem, we consider 2 cases, \_\_\_\_\_\_\_

Or \_\_\_\_\_\_\_\_.

* Median, Mode
* Recursive, Iterative
* **Leave object, Take object  page 93**
* Sequentially. Parallel

Random Access Machine (RAM) can execute \_\_\_\_\_\_\_\_\_ instructions

* only logical
* parallel
* only arithmetic
* **logical and arithmetic**

Using \_\_\_\_\_\_\_ algorithm, efficiency is not given much importance

* Greedy
* Merge sort
* **Processing**
* Brute Force

Bubble sort takes theta \_\_\_\_\_\_\_\_\_ in the worst case

* **(n2)                    page 39**
* (n)
* log(n)
* nlog(n)

If matrix A of dimension p × q is multiply with matrix B of dimension q × r, then dimension of resultant matrix is:

* q × r
* r × p
* **P x r**
* P x q

Dynamic Programing algorithms often use some kind of \_\_\_\_\_\_\_\_ to store the results of intermediate sub-problems

* variable
* stack
* **Table**
* loop

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is in-place sorting algorithm.

* **Bubble sort             (Page 54)**
* Merge sort
* Linear search
* Binary Search

Which one of the following problems can be solved using dynamic problem?

* Bubble sort problem
* **Matrix chain multiplication problem      page 85**
* Greedy search problem
* Fractional knapsack problem

In chain matrix multiplication, solutions of the sub-problems are stored in a

\_\_\_\_\_\_\_\_\_.

* Array
* **Table               page 86**
* Tree
* Link list

What is the average running time of a quick sort algorithm?

* O(n^2)
* O(n)
* **O(n log n) (Page 49)**
* O(log n)

Sorting Algorithms having O \_\_\_\_\_\_\_ running time are considered to be slow ones.

* (n)
* **(n^2)            (Page 39)**
* (nlog(n))
* (log(n))

While solving Selection problem, in Sieve technique we partition input data

\_\_\_\_\_\_\_\_

* In increasing order
* In decreasing order
* **According to Pivot**
* Randomly

\_\_\_\_\_\_\_\_ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.

* Loop
  + **Memoization                   page 74**
* Recursion
* Function

In average-case time the probability of seeing input is denoted by \_\_\_\_\_\_\_.

* p{I}
* p[I]
* p<i>
* **p(i)                                 page 13**

While applying the Sieve technique to selection sort, how to choose a pivot element.

* Through mean
* Linear
* **Randomly                   page 35**
* Sequentially

Number of \_\_\_\_\_\_\_ of the pseudo code are counted to measure the running time.

* Inputs
* Outputs
* **Steps              page 13**
* Pages

Developing a dynamic programming algorithm generally involves \_\_\_\_\_\_

separate steps.

* One
* **Two                             page 75**
* Three
* Four

8n^2+2n+3 will exceed c28(n), no matter how large we make \_\_\_\_\_.

* n
* 2n
* **c2                           page 25**
* this quadratic equation

The running time of quick sort algorithm\_\_\_\_\_\_\_\_\_.

* Is impossible to compute
* Has nothing to do with pivot selection
* Is Random upon each execution
* **Greatly influenced by the selection of pivot    page 49**

\_\_\_\_\_\_\_\_\_ involves breaking up the problem into sub problems whose solutions can be combined to solve the global problem.

* Complexity Theory
* Dynamic programming solution
* **Divide and Conquer Strategy          page 34**
* Greedy Algorithms

In \_\_\_\_\_\_\_\_\_\_\_\_\_ we have to find rank of an element from given input.

* Merge sort algorithm
* **Selection problem                 page 34**
* Brute force technique
* Plane Sweep algorithm

How many steps are involved to design the dynamic programming strategy?

* 2
* 3
* 1
* **4                                     page 92**

In Bucket sort, if there are duplicates then each bin can be replaced by a

* **Linked list         page 69**
* Hash table
  + Stack
* Heap

In merge sort algorithm, we split the array \_\_\_\_\_\_ to find index q.

* from end
* from start
* **midway   page 28**
* both from start or end

Find the maximum value of the items which can carry using knapsack Knapsack weight capacity = 50.

Item  Weight Value

11070

22020

33080

470  200

* 280
* 100
* 90
* **200**

In 2-d maxima problem a point p is said to be dominated by point q if

\_\_\_\_\_\_\_\_\_.

* p.x <= q.x
* **bp.x <= q.x and p.y <= q.y                   page 17**
* p.y <= q.y
* p.x >= q.x and p.y >=q.y

Sorting can be in \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

* Increasing order only
* Decreasing order only
* **Both increasing and decreasing order**
* Random order

Recurrence can be described in terms of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

* Array
* Linear
* **Tree                                  page 31**
* Graph

The brute-force algorithm for 2D-Maxima runs in order O(\_\_) time.

* n
* n(log n)
* **n\*n                            page 18**
* n3

In plane sweep approach of solving geometric problems, a \_\_\_\_\_\_\_\_\_ is swept across the plane.

* **Line                        page 18**
* Plane
* Cube
* Box

Which of the following is calculated with Big Omega notation?

1. Medium bounds

* Upper bounds

c. Lower bounds                     Page – 25

* Both upper and lower bounds

\_\_\_\_\_\_\_\_\_ is always based on divide and conquer strategy.

* Bubble sort
* Selection sort
* Pigeon sort
* **Quick sort                 page 46**

If a matrix has three rows and two columns, then dimensions of matrix will be:

* **3×2**
* 2×3
* 3×3
* 2×2

Asymptotic notations are used to describe \_\_\_\_\_\_\_ of an algorithm.

* Length
* **running time                  google**
* size
* compile time

Catalan numbers are related the number of different \_\_\_\_\_\_ on ‘n’ nodes.

* Arrays
* linked lists
* **binary trees            page 85**
* functions

Applying the sieve technique to selection problem, \_\_\_\_\_\_\_\_ element is picked from array.

* Output
* Total
* Input
* **Pivot                page 35**

Dynamic Programming approach is usually useful in solving \_\_\_\_\_\_\_

problems.

* **Optimization                       google**
* Array
* Normal
* Loop

In recursive formulation of knapsack Problem: V [0, j] = \_\_\_\_\_\_\_\_ for j>=0

* -1
* **0                 page 93**
* 1
* 2

\_\_\_\_\_\_\_\_ is a linear time sorting algorithm.

* Merge sort
* **Radix sort                          page 71**
* Quick sort
* Bubble sort

Quick sort is one of the \_\_\_\_\_ sorting algorithm.

* **Fastest         page 19**
* Slowest
* Major
* Average

The time assumed for each basic operation to execute on RAM model of computation is \_\_\_\_\_.

* Infinite
* Continuous
* **Constant                    page 10**
* Variable

In Sieve Technique, we know the item of interest.

* True
* **False**

While analyzing algorithms, \_\_\_\_\_\_\_ and \_\_\_\_\_\_\_ usually considered difficult to calculate.

* **Floor, ceiling       google**
* Row, Column
  + Finite, Infinite
* Graph, Tree

While analysis of the brute-force maxima algorithm, an array sorted in the reverse order is the type of \_\_\_\_\_\_\_\_\_ case input.

* Best
* **Worst              page 14**
* Somewhat bad
* Average

\_\_\_\_\_\_\_\_\_ is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

* Mean
* Mode
* Average
* **Median         page 34**

In asymptotical analysis of n(n-3) and 4n\*n, as n becomes large, the dominant (fastest growing) term is some constant times \_\_\_\_\_\_\_.

* n+1
* n-1
* n
* **n\*n                     page 23**

In addition to passing in the array itself to Merge Sort algorithm, we will pass in other arguments which are indices.

* Three
* **Two**
* Four
* Five

In 2d-maximal problem, a point is said to be if it is not dominated by any other point in that space.

* Member
* Minimal
  + **Maximal**
* Joint

Counting sort assumes that the numbers to be sorted are in the range

\_\_\_\_\_\_\_\_.

1. K to n where n is large

* K to n where k is small
  + **1 to k where k is small**
* k to n where n is small

Insertion sort is an efficient algorithm for sorting a \_\_\_\_\_\_\_\_\_\_ number of elements

* Small
* Extra large
  + **Large**
* Medium

If the indices passed to merge sort algorithm are \_\_\_\_\_\_\_\_\_ then this means that there is only one element to sort.

* **Small                                page 28**
* Large
* Equal
* Not Equal

In Knapsack Problem, each item must be entirely accepted or rejected, is called \_\_\_\_\_\_ problem.

* Linear
* Fractional
* **0-1**
* Optimal

If the time complexity of an algorithm is O(n). then it is called \_\_\_\_\_\_\_ time complexity.

* **Linear                                                            Wikipedia**
* Constant
* Average
* Exponential

In the case of \_\_\_\_\_\_\_\_\_ analysis does not depend upon on the distribution of input.

* Merge sort
* Insertion sort
  + **Quick Sort**
* Heap sort

We can use the \_\_\_\_\_\_\_\_\_\_\_ Property to devise a recursive formulation of the edit distance problem.

* Small substructure
* Algorithmic
* Real
* **Optimal substructure                         page 78**

The following sequence is called \_\_\_\_\_\_\_\_\_\_\_\_

* 1,2,3,5,8,13,21,34,55,…..
  + **Fibonacci sequence                   page 73**
* Optimal sequence
  + Optimize Sequence
* Overlapping sequence

Which one sorting algorithm is best suited to sort an array of 2 million elements?

* Bubble sort
* Insert sort
* Merge sort
* Quick sort
* **Ridx Sort    page 71**

We can improve the performance of quick sort if we could be able to \_,\_\_\_\_\_\_\_\_\_\_.

* **Select two or more pivots              page 34**
* Skip any sub-array completely
  + Skit Input elements somehow
* Eliminate recursive calls

The problem with the brute-force algorithm is that is uses \_\_\_\_\_\_\_\_ in pruning out de

* Worst-case time
* **No intelligence                        page 18**
* Outside looping
* Artificial intelligence

In chain matrix multiplication, the order of the matrices \_\_\_\_\_\_\_\_\_\_.

* Can be changed
* **Can not be changed                      page 85**
* is equal
* is reverse

In quick sort algorithm, we choose pivot\_\_\_\_\_\_\_\_\_\_\_.

* Always the smallest element
* Greater than 5
* **Randomly                  page 35**
* Less than 5
* In Heap Sort algorithm. Heapify procedure is \_\_\_\_\_\_\_\_ in nature.
* Recursive
* **Non-Recursive                 page 43**
* Fast
* Slow

When matrix A of 5x 3 is multiplied with matrix B of 3 x 4 then the number of multiplications required will be \_\_\_\_\_\_\_\_\_\_\_.

* 15
* 12
* 36
* **60**

An algorithm is said to be correct if for every \_\_\_\_\_\_ instance, it halts with the correct \_\_\_\_\_\_.

* **Input, Output                       page 13**
* Design, Analysis
* Value, Key
* Key, Analysis

In chain matrix multiplication, table is filled \_\_\_\_\_\_\_\_\_ to find the multiplication of matrix.

* row wise
* column wise
* diagonally
* **bottom-to-up         page 86**

If we have an equation 8n2+7f\*n + 5f + 6 then is large, \_\_\_\_\_\_\_\_ term will be muchxxxxxxxthe n term and will dominate the running time.

* f g (n)
* g (n) \* 2
* **n \* 2                  page 23**
* f (n)

For quick sort algorithm. Partitioning takes theta \_\_\_\_\_\_\_\_.

* (n)
* log(n)
* n log (n)
* **n2log (n)**

In Heap Sort algorithm, the maximum levels an element can move upward is

\_\_\_\_\_\_\_

* **Theta (log n)          page 43**
* Big-ch (log n)
* Omega (log n)
* 0 (1) i.e. Constant time

\_\_\_\_\_\_\_ programming is essentially recursion without repetition.

* Array
* Fast
* **Dynamic**
* n (log n)

There are no hard formal rules to the syntax of the \_\_\_\_\_\_\_\_ code.

1. Basic

* Programming
* Pseudo
* Assembly

In Heap Sort algorithm, to remove the maximum element every time.

* We call Build-Heap procedure
* Heap Sort algorithm terminates without result
* We call heapify procedure
* Nothing happens

Which process is used for avoiding unnecessary repetitions and looking them up again if we need them later.

* Greedy Approach
* **Memoization                      page 74**
* Divide and conquer
* Recursion

The worst-case running time of Quick sort is \_\_\_\_\_\_\_\_\_ in order to sort an array of n element.

* **O(n log n)                       page 49**
* O(n)
* O(n2)
  + O(log n)

Boolean operation is a \_\_\_\_\_\_\_\_\_ operation on an idealized RAM model of computation.

* Advance
* String
* **Basic**
* Normal

In chain matrix multiplication, if there are n items, there are \_\_\_\_\_\_\_\_ ways in which outer most pair of parentheses can placed.

* n^2
* 2n
* n+1
* **d. n-1           page 85**

The number of nodes in a complete binary tree of height h is:

* \* (h+1) – 1
* \* (h+1)
* **b. 2^(h+1) – 1                             page 40**
* ((h+1)^2) – 1
* Heaps can be stored in arrays without using any pointers; this is due to the \_\_\_\_\_\_\_\_\_\_\_\_ nature of the binary tree,   
  [**left-complete**](http://www.vuzs.info/)  
  right-complete  
  tree nodes  
  tree leaves  
    
  Sieve Technique can be applied to selection problem?   
  [**True**](http://www.vuzs.info/)  
  False  
    
  A heap is a left-complete binary tree that conforms to the \_\_\_\_\_\_\_\_\_\_\_   
  increasing order only  
  decreasing order only  
  [**heap order**](http://www.vuzs.info/)  
  (log n) order  
    
  A (an) \_\_\_\_\_\_\_\_\_ is a left-complete binary tree that conforms to the heap order   
  [**heap**](http://www.vuzs.info/)  
  binary tree  
  binary search tree  
  array  
    
  Divide-and-conquer as breaking the problem into a small number of   
  pivot  
  Sieve  
  [**smaller sub problems**](http://www.vuzs.info/)  
  Selection  
    
  In Sieve Technique we do not know which item is of interest   
  [**True**](http://www.vuzs.info/)  
  False  
    
  The recurrence relation of Tower of Hanoi is given below T(n)={1 if n=1 and 2T(n-1) if n >1 In order to move a tower of 5 rings from one peg to another, how many ring moves are required?   
  16  
  10  
  [**32**](http://www.vuzs.info/)31   
    
  In the analysis of Selection algorithm, we eliminate a constant fraction of the array with each phase; we get the convergent \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ series in the analysis,   
  linear  
  arithmetic  
  [**geometric**](http://www.vuzs.info/)  
  exponent  
    
  For the heap sort, access to nodes involves simple \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operations.   
  [**arithmetic**](http://www.vuzs.info/)  
  binary  
  algebraic  
  logarithmic   
    
  For the sieve technique we solve the problem,  
  [**recursively**](http://www.vuzs.info/)mathematically  
  precisely  
  accurately  
    
  The sieve technique works in \_\_\_\_\_\_\_\_\_\_\_ as follows  
  [**phases**](http://www.vuzs.info/)numbers  
  integers  
  routines  
    
  Slow sorting algorithms run in,  
  [**T(n^2)**](http://www.vuzs.info/)T(n)  
  T( log n)  
    
  A (an) \_\_\_\_\_\_\_\_\_ is a left-complete binary tree that conforms to the heap order  
  [**heap**](http://www.vuzs.info/)  
  binary tree  
  binary search tree  
  array  
    
  In the analysis of Selection algorithm, we make a number of passes, in fact it could be as many as,  
  T(n)  
  [**T(n / 2)**](http://www.vuzs.info/)  
  log n  
  n / 2 + n / 4  
    
  The sieve technique is a special case, where the number of sub problems is just  
  5  
  many  
  [**1**](http://www.vuzs.info/)  
  few  
    
  In which order we can sort?  
  increasing order only  
  decreasing order only  
  [**increasing order or decreasing order**](http://www.vuzs.info/)  
  both at the same time  
    
  Analysis of Selection algorithm ends up with,  
  T(n)  
  T(1 / 1 + n)  
  T(n / 2)  
  [**T((n / 2) + n)**](http://www.vuzs.info/)  
    
  The analysis of Selection algorithm shows the total running time is indeed \_\_\_\_\_\_\_\_in n,  
  arithmetic   
  geometric   
  [**linea**](http://www.vuzs.info/)**r**orthogonal   
    
  How many elements do we eliminate in each time for the Analysis of Selection algorithm?   
  [**n / 2 elements**](http://www.vuzs.info/)(n / 2) + n elements   
  n / 4 elements   
  2 n elements   
    
  For the heap sort we store the tree nodes in   
  [**level-order traversal**](http://www.vuzs.info/)   
  in-order traversal   
  pre-order traversal   
  post-order traversal
* One of the clever aspects of heaps is that they can be stored in arrays without using any \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.   
  [**pointers**](http://www.vuzs.info/)constants  
  variables  
  functions
* Divide-and-conquer as breaking the problem into a small number of   
  pivot  
  Sieve  
  [**smaller sub problems**](http://www.vuzs.info/)Selection  
    
  How much time merge sort takes for an array of numbers?   
  T(n^2)  
  T(n)  
  T( log n)  
  [**T(n log n)**](http://www.vuzs.info/)  
    
  The reason for introducing Sieve Technique algorithm is that it illustrates a very important special case of,   
  [**divide-and-conquer**](http://www.vuzs.info/)decrease and conquer  
  greedy nature  
  2-dimension Maxima
* The number of nodes in a complete binary tree of height h is  
  [**2^(h+1) – 1**](http://www.vuzs.info/)  
  2 \* (h+1) – 1  
  2 \* (h+1)  
  ((h+1) ^ 2) – 1
* Sieve Technique applies to problems where we are interested in finding a single item from a larger set of \_\_\_\_\_\_\_\_\_\_\_\_\_  
  [**n items**](http://www.vuzs.info/)  
  phases  
  pointers  
  constant
* Memorization is?  
  To store previous results for future use  
  [**To avoid this unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later**](http://www.vuzs.info/)To make the process accurate  
  None of the above
* Which sorting algorithm is faster  
  O (n log n)  
  O n^2  
  [**O (n+k)**](http://www.vuzs.info/)O n^3
* Quick sort is  
  Stable & in place  
  [**Not stable but in place**](http://www.vuzs.info/)Stable but not in place  
  Some time stable & some times in place
* One example of in place but not stable algorithm is  
  Merger Sort  
  [**Quick Sort**](http://www.vuzs.info/)Continuation Sort  
  Bubble Sort
* In Quick Sort Constants hidden in T(n log n) are  
  Large  
  Medium  
  [**Small**](http://www.vuzs.info/)Not Known
* Continuation sort is suitable to sort the elements in range 1 to k  
  K is Large  
  K is not known  
  K may be small or large  
  [**K is small**](http://www.vuzs.info/)
* In stable sorting algorithm.  
  [**If duplicate elements remain in the same relative position after sorting**](http://www.vuzs.info/)One array is used  
  More than one arrays are required  
  Duplicating elements not handled
* Which may be a stable sort?  
  Merger  
  Insertion  
  Both above  
  [**None of the above**](http://www.vuzs.info/)
* An in place sorting algorithm is one that uses \_\_\_ arrays for storage  
  Two dimensional arrays  
  More than one array  
  [**No Additional Array**](http://www.vuzs.info/)None of the above
* We do sorting to,  
  keep elements in random positions  
  keep the algorithm run in linear order  
  keep the algorithm run in (log n) order  
  [**keep elements in increasing or decreasing order**](http://www.vuzs.info/)
* In Sieve Technique we donot know which item is of interest  
  [**True**](http://www.vuzs.info/)False
* Sorting is one of the few problems where provable \_\_\_\_\_\_\_\_ bonds exits on how fast we can sort,  
  upper  
  [**lower**](http://www.vuzs.info/)average  
  log n
* The reason for introducing Sieve Technique algorithm is that it illustrates a very important special case of,  
  [**divide-and-conquer**](http://www.vuzs.info/)decrease and conquer  
  greedy nature  
  2-dimension Maxima
* Analysis of Selection algorithm ends up with,  
  [**T(n)**](http://www.vuzs.info/)T(1 / 1 + n)  
  T(n / 2)  
  T((n / 2) + n)
* In in-place sorting algorithm is one that uses arrays for storage :  
   An additional array  
  [**No additioanal array**](http://www.vuzs.info/)  
  Both of above may be true according to algorithm  
  More than 3 arrays of one dimension.
* Which sorting algorithn is faster :  
  O(n^2)  
  [**O(nlogn)**](http://www.vuzs.info/)  
  O(n+k)  
  O(n^3)
* The running time of quick sort depends heavily on the selection of  
  No of inputs  
  Arrangement of elements in array  
  Size o elements  
  [**Pivot elements**](http://www.vuzs.info/)Which may be stable sort:  
  Bubble sort  
  Insertion sort  
  [**Both of above**](http://www.vuzs.info/)  
  For the Sieve Technique we take time  
  [**T(nk)**](http://www.vuzs.info/)T(n / 3)  
  n^2  
  n/3