Q1. Study updated code of binary search

// read about random array of 1000 nos generated and how binary search is applied

Q2. Study the hash functions given ppt of hash table

Q3. Implement parentheses balancing problem using HashMap and Stack

// see the algorithm in ppt of stack (day1)

Q4. apply selection and insertion sort on array of characters

Q5. apply quick sort on array of characters

// study updated code of quick sort

—------------------------------------------------------------------------------------

Q6. Draw tree of mergesort and quicksort for input array {10,4,7,3,2,5}

// Extra

Write a function SLL class to insert element at right position

Ex. if SLL is 10 -> 67 -> 89 -> null

And 50 needs to be inserted then put 50 before 67

So o/p SLL will be

10 -> 50 -> 67 -> 89 -> null

// Extra

Apply quicksort to sort array of strings based on first alphabet of string

If two strings with same first alphabet then keep them random

i/p : { “iacsd” , “cdac” , “pune”}

o/p : {“cdac” , “iacsd” , “pune” }

i/p : { “mba” , “maharashtra”, “canteen” }

i/p : { “canteen” , “mba” , “maharashtra”}