

9:9

array [10, 5, 3, 9, 22, 24, 28, 27, ?]

a) select value & mention all possible values.

any values from 11 to 21 will make a quick sort perform efficient,

this is because by taking  
no. ~~of~~ # between 11 to 21  
the first q.v. index will  
not ~~perform~~ be able to perform  
worse case.

we found these no. mostly  
or ~~say~~ almost equal to middle  
value, so the quick sort  
become fast and efficient

11  
a)



b)

10, 5, 3, 9, 22, 24, 28, 27, 12  
10, 5, 3, 9, 22, 24, 28, 27, 12  
10, 5, 3, 9, 12, 24, 28, 27, 22  
3, 5, 9, 10, 12, 24, 28, 27, 22  
3, 5, 9, 10, 12, 24, 28, 27, 22  
3, 5, 9, 10, 12, 24, 28, 27, 22  
3, 5, 9, 10, 12, 24, 28, 27, 22  
3, 5, 9, 10, 12, 22, 28, 27, 24  
3, 5, 9, 10, 12, 22, 28, 27, 24  
3, 5, 9, 10, 12, 22, 24, 27, 28  
3, 5, 9, 10, 12, 22, 24, 27, 28

Sorted



Q. 10

## Divide & Conquer

Divide & Conquer work with dividing the problem into sub-problems, conquer each sub problem recursively and combine these solutions.

It is recursive.

I would not treat them as something completely diff. Because they both work by recursively breaking down a problem into two or more sub problems.

It is top down approach. In this subproblems are independent of each other. (i.e) merge sort.

## Dynamic Programming

It is a technique for solving problems with overlapping subproblems. Each sub-prob is solved only once and result of each sub prob is stored in a table (generally we implement array or hash table) for future references. and technique for storing the sub-prob solution is known as memorization.

It is non-recursive

It solve prob 1 and reuse it. In this problems are interdependent (i.e) matrix multiplication