Question 1)

Can not solve.

Question 2)

This algorithm takes exponent and number as input. Power function calculates the result. Function checks if the exponent is zero. If it is, it returns 1. If the exponent is not zero, the function calculates the result by calling itself with num and exponent//2 as arguments. // this means that calculate this operation as integer not float number.After the recursive call, the function checks if the exponent is even. If it is, it returns the result of the recursive call multiplied by itself. This is equivalent to squaring the result of the recursive call, which is the same as taking power of number by two.If exponent is not even, This means that we must multiply the square of result by number one more.

Worst Case: This algorithm always divides operations on every recursive calls.So in worst case it is still O(log n).

Question 3)

In this algorithm, the function takes a 9x9 sudoku as input. If the value on sudoku[i][j]=0 this means that this entry is empty and it must be replaced with input.But function must find valid input fort his place.So it calls isValid function for control input if it can be replaced by that number.First it control the row if same number is in the row, if it can not find, it looks for column if same number is in the column.If it can not found it finally looks 3x3 part of sudoku.If it is still can not found the input, then this means that this input can be replaced to there.It does this operations until solve sudoku.

Worst case:This algorithm has 3 for loops and every of them Works 9 time.It is O(9^3).

Question 4)

With insertion sort:

{6, 8, 9, 8, 3, 3, 12} 8 is bigger than 6 no change.

{6, 8, 9, 8, 3, 3, 12} 9 is bigger than 8 no change.

{6, 8, 8, 9, 3, 3, 12} 8 is smaller than 9 change.

{6, 8, 8, 3, 9, 3, 12} 3 is smaller than 9 change

{6, 8, 3, 8, 9, 3, 12} 3 is smaller then 8 change.

{6, 3, 8, 8, 9, 3, 12} 3 is smaller than 8 change.

{3, 6, 8, 8, 9, 3, 12} 3 is smaller than 6 change.

{3, 6, 8, 8, 3, 9, 12} 3 is smaller than 9 change

{3, 6, 8, 3, 8, 9, 12} 3 is smaller then 8 change.

{3, 6, 3, 8, 8, 9, 12} 3 is smaller then 8 change.

{3, 3, 6, 8, 8, 9, 12} 3 is smaller than 6 change.

array is sorted.

With quick sort:

{6, 8, 9, 8, 3, 3, 12} Select middle element 8 pivot.

{6, 8, 9, 12, 3, 3, 8} Move it to the end.

{6, 8, 9, 12, 3, 3, 8} Search element on the left which is greater or equal to 8.

{6, 8, 9, 12, 3, 3, 8} We found 8 as second element.So we search an element from right to left that is smaller than pivot.Then we change their positions.

{6, 3, 9, 12, 3, 8, 8} We continue search from left to find a value that is equal or greater than pivot.We found 9.Then we search right to left that is smaller than pivot we found 3.Swap them.

{6, 3, 3, 12, 9, 8, 8} We reach middle element it is bound.Return pivot to the position. Left side is smaller than pivot and right side is bigger than.We take them as sub arrays.

{6, 3, 3, 8, 9, 8, 12} {6,3} array.We control and change 3 and 6 position.

{3, 3, 6, 8, 9, 8, 12} {3,6} control.

{3, 3, 6, 8, 9, 8, 12} left 6.It is sorted.

{3, 3, 6, 8, 9, 8, 12} Then for the right sub tree 9,12,8.

{3, 3, 6, 8, 9, 12, 8} We chose middle 8 as pivot and take it to the end.

{3, 3, 6, 8, 9, 12, 8} Made controls and can not found number smaller then pivot.So change pivot and left number.

{3, 3, 6, 8, 8, 12, 9} {12,9} can not found.So change pivot and left number.

{3, 3, 6, 8, 8, 9, 12}

array is sorted.

With buble sort:

{6, 8, 9, 8, 3, 3, 12} Compare numbers two by two and swap them if right is smaller.

{6, 8, 8, 9, 3, 3, 12} Change 9 and 8. Since 8<9

{6, 8, 8, 9, 3, 3, 12} Change 9 and 3, Since 3<9

{6, 8, 8, 3, 9, 3, 12} Change 9 and 3, Since 3<9

{6, 8, 8, 3, 3, 9, 12} Change 8 and 3, Since 3<8

{6, 8, 3, 8, 3, 9, 12} Change 8 and 3, Since 3<8

{6, 8, 3, 3, 8, 9, 12} Change 8 and 3, Since 3<8

{6, 8, 3, 3, 8, 9, 12} Change 8 and 3, Since 3<8

{6, 3, 8, 3, 8, 9, 12} Change 8 and 3, Since 3<8

{6, 3, 3, 8, 8, 9, 12} Change 6 and 3, Since 3<6

{3, 6, 3, 8, 8, 9, 12} Change 6 and 3, Since 3<6

A stable sorting algorithm maintains the relative order of the items with equal sort keys. An unstable sorting algorithm does not. In other words, when a collection is sorted with a stable sorting algorithm, items with the same sort keys preserve their order after the collection is sorted.

If array has equal numbers in it, In quick sort tis causes unstability.In this array we have 2 3 and 2 8 and this makes quick sort unstable.But insertion and buble sort still is stable.

Question 5)

a) Exhaustive search is a brute force approach with some special aproaches like,

-create all possible solutions in a systematic manner

- evaluate potential solutions one by one, remove infeasible ones and

keep track of the best one found so far

- when there is no more to evaluate, output the solution found.

b)In Ceasers Chipher key is shifted with all characters.With brute force attacks hacker only has to shift chipper-text to shift with some number.So it is vulnerable to brute force attacks.

AES is vulnerable to brute force attacks because the encryption key is usually relatively short (e.g., 128 bits, 192 bits, or 256 bits) and there are only a limited number of possible keys. Another reason is AES vulnerable to brute force attacks is that the algorithm uses the same key for both encrypting and decrypting the text. This

means that if the attacker is able to find the correct key, they can use it to both encrypt and decrypt the ciphertext, effectively breaking the encryption.

c)In primality testing, if a>0 and b>1 and n = a^b so n is not an primal number.If n < a^b and there is not any divisible, then this number is primal.This is trial division method. For example, if the number being tested has 100 digits, the trial division method would have to divide it by all potential divisors up to the square root of the number, which could be a very large number with hundreds of digits..So In bigger numbers It increases more much with repedeatly.