1. **Write short note on optimization:**

Finding an alternative with the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones. In comparison, maximization mean trying to attain the highest or maximum result or outcome without regard to cost or expense. Practice of optimization is restricted by the lack of full information, and the of time to evaluate what information is available. In computer simulation of business problems, optimization is achieved usually by using linear programming techniques of operations research.

1. **What are the different algorithms you know**

I know different type of algorithms which is the fundamental type of algorithms.

Insertion sort algorithms, Selection sort algorithms, Recursive algorithms, Dynamic algorithms, Greedy algorithms.

**Recursive Algorithm:** A Recursive algorithm is an algorithm which calls itself with smaller input values and which obtain the result for the current input by applying simple operations to the returned value for the smaller input. Generation of Fectorial, Fibonacci number series are the examples of Recursive algorithm.

**Greedy Algorithms:** A greedy algorithm is a simple, intuitive algorithm that is used in optimization problems. The algorithm makes the optimal choice at each step as it attempts to find the overall optimal way to solve the entire problem.

**Dynamic Programming Algorithms:** Dynamic Programming is also used in optimization problems. Like divide-and-conquer method, Dynamic Programming solves problems by combining the solutions of subproblems. Moreover, Dynamic Programming algorithm solves each sub-problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time.

1. **Why are you learning so many algorithms**

We learning so many algorithms to use those algorithms to give more efficient solution for complex problem we do everyday.

Learning algorithms helps me to speed up my thinking ability for other programming problems.

1. **Show analysis of a recursive algorithm**

Factorial(n)

{

If(n=0)

Return 1;

Else

n\*factorial(n-1)

}

**Time complixity analysis:**

Let’s assume ==,\*,-, operatos’s cost constant 1 time.

T(n)= T(n-1)+1+1+1

=T(n-1)+3

=T(n-2)+6

=T(n-3)+8

=T(n-k)+3k

T(0)=1

Interms of T(0)

n-k=0

n=k

T(n)=T(0)+3n

T(n)=3n+1

The Time complexity in the worst case: O(n)