Dynamic Programming:

Dynamic Programming is an optimization over recursion. Whenever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

For example, if we write simple recursive solution for Fibonacci Numbers, we get exponential time complexity and if we optimize it by storing solutions to subproblems, time complexity reduces to linear

```
int fib(int n)
{ if (n <= 1)
      return n;
  return fib(n-1) + fib(n-2);
}
```

Recursion: Exponential

Dynamic Programming : Linear

```
f[0] = 0;
f[1] = 1;
for (i=2 ; i<=n ; i++)
{ f[i] = f[i-1] + f[i-2];
}
return f[n];
```