

## Pre-requisites

- Recursion

## Dynamic Programming

---

Dynamic programming is a method for solving a complex problem by breaking it down into a collection of simpler subproblems, solving each of those subproblems just once, and storing their solutions. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution, thereby saving computation time.

"overlapping subproblems" - A problem has overlapping subproblems if finding its solution involves solving the same subproblem multiple times.

Approaches to solve a problem using dynamic programming:

1. Top-down approach: Memoization
  - Start solving the problem by solving the largest subproblem first and then use the solution of the largest subproblem to solve the next subproblem.
2. Bottom-up approach: Tabulation
  - Start solving the problem by solving the simplest subproblem first and then use the solution of the simplest subproblem to solve the next subproblem.

Example: Factorial of a number 'n'

Factorial of a number n is the product of all positive integers less than or equal to n.

Factorial of 5 is  $5 \times 4 \times 3 \times 2 \times 1 = 120$ . Factorial of 7 is  $7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$ .

Algorithm: Bottom-up approach

1. Start from the smallest subproblem.

$dp[0] = 1$   $dp[1] = 1$   $dp[2] = 2$

2. Use the solution of the smallest subproblem to solve the next subproblem.

$dp[3] = dp[2] * 3 = 6$

3. Repeat step 2 until the solution of the largest subproblem (n) is found.
4. return  $dp[n]$  which is the solution of the problem.

$F(0) = 1$

$F(1) = 1$   $F(2) = F(1) * 2 = 2$   $F(3) = F(2) * 3 = 6$   $F(4) = F(3) * 4 = 24$   $F(5) = F(4) * 5 = 120$   $F(6) = F(5) * 6 = 720$

Recursion:

Recursion is a phenomenon in which a function calls itself.

Question:

Kumar is interested in investing in CP Inc after hearing of their profits. He believes that CP Inc will make more money in the future . To confirm this he uses his secret time travelling machine to travel N days into the future. Once in the future he obtains a record of the prices of a single stock for each of the N days and returns to his timeline. After he returns he decides to analyze how to make the most profit by investing . Given a list of prices of the stock for N days , help Kumar find the maximum profit he can make by buying one stock and selling one stock.

Note : Assume Kumar has an infinite amount of money , so he can buy any stock of any price . Your goal is to maximize his profit. Kumar can also refuse to buy or sell any stock , in that case his profit is 0. Also note that kumar has to first buy a stock before he can sell the same stock

Input Description: The first line contains a single integer N denoting the number of days for which stock prices are known. The second line contains N integers denoting the prices of the stock. The ith integer is the price of the stock on the ith day ( $1 \leq i \leq 1000$ )

Output Description: The output must consist of a single integer : the maximum profit

Sample Input : 4 1 2 3 4 Sample Output : 3