Dynamic programming

Suppose a array is given and asked to find whether the sum exist in any subset .

Approach

A = [2,5,3,4] , number = 12

Create a array of size M and check for each element from right to left.

First element is 2 w’ll check if a[12] is set is set then continue

Also if a[12 – 2] is set a[12] to 1

Else continue .

This will run till from m to the element 12-2

Similarly for 5 12-5

Similarly for 3 12-3

So on

To find the elements of subset used in making the sum , just replace 1 with n-1th element.

Max and min in two dimension grid

Given a 2-d array of dimension NxM we are position (1,1) , the array is filled with random values , find the min value to reach from 1,1 ->n,m .And we can move right and down.

Approach go reverse

A[n][m] is known we can calculate a[n][m-1] and a[n-1[m] so on…

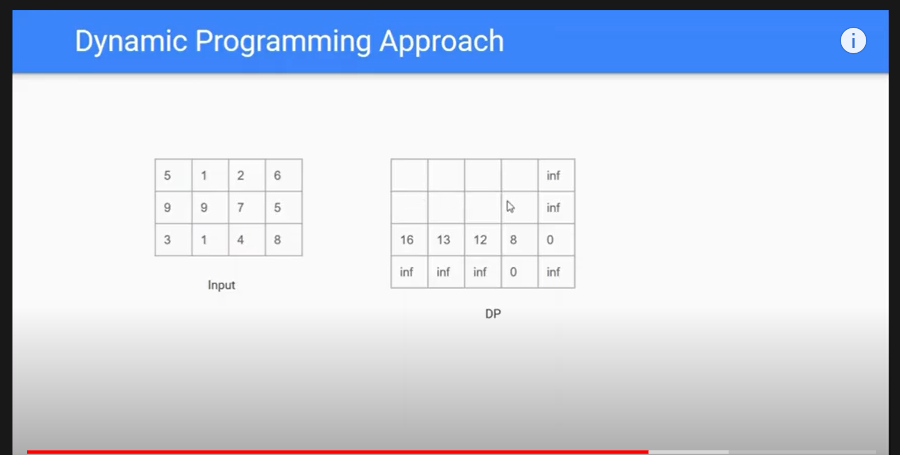
But we can go like that so we will take dp array of n+1 and m+1 size.

And fill dp[n+1][1tom+1] = inf except dp[n+1][m] = 0;

And fill dp[1 to n+1][m+1] = inf except dp[n][m+1] = 0;

Move across from dp[n][m] to dp[n][1tom] and find minimum value.

Using lower row value find the value of current row.



**Longest Substring problem**

Time complexity O(n square)

Important observation if a string of length n is palindrome then substring of length n-2 i.e leaving first and last index is also a palindrome.

Declare 2d array with rows define the length of palindrome string and column defining the ending position of palindrome substring and mark it 1 if it palindrome or 0

0th row is empty string so it is by default palindrome and row 1 is also palindrome of length 1

Next all row will be checked with 2 condition if first and last character matches and if dp[row-2][column-1] ==1 if so then it is also a palindrome.

Explanation for second condition we are doing row -2 because we are checking for n-2 length substring and that will end at j-1th position



**Finding different possible ways**

Consider nxm matrix ,Find number of ways we can reach we can reach nxm from 1x1 if we are allowed to move right and down .

Approach:

From [n][1-m] and [1-n][m] we can reach there in 1 way.

Now in n-1th row consider [n-1][m-1] block we can reach there in [n-1][m] + [n][m-1] ways…

Like wise we can fill the whole row and move upwords.