Dynamic Programming | Set 14 (Maximum Sum Increasing Subsequence)

Given an array of n positive integers. Write a program to find the sum of maximum sum subsequence of the given array such that the intgers in the subsequence are sorted in increasing order. For example, if input is {1, 101, 2, 3, 100, 4, 5}, then output should be 106 (1 + 2 + 3 + 100), if the input array is {3, 4, 5, 10}, then output should be 22 (3 + 4 + 5 + 10) and if the input array is {10, 5, 4, 3}, then output should be 10

Solution
This problem is a variation of standard Longest Increasing Subsequence (LIS) problem. We need a slight change in the Dynamic Programming solution of LIS problem. All we need to change is to use sum as a criteria instead of length of increasing subsequence.

Following is C implementation for Dynamic Programming solution of the problem.

```
/* Dynamic Programming implementation of Maximum Sum Increasing
   Subsequence (MSIS) problem */
#include<stdio.h>

/* maxSumIS() returns the maximum sum of increasing subsequence in arr[] of
   size n */
int maxSumIS( int arr[], int n )
{
   int *msis, i, j, max = 0;
   msis = (int*) malloc ( sizeof( int ) * n );

   /* Initialize msis values for all indexes */
   for ( i = 0; i < n; i++ )
      msis[i] = arr[i];

   /* Compute maximum sum values in bottom up manner */
   for ( i = 1; i < n; i++ )
      for ( j = 0; j < i; j++ )
         if ( arr[i] > arr[j] && msis[i] < msis[j] + arr[i])
            msis[i] = msis[j] + arr[i];

   /* Pick maximum of all msis values */
   for ( i = 0; i < n; i++ )
      if ( max < msis[i] )
         max = msis[i];

   /* Free memory to avoid memory leak */
   free( msis );

   return max;
}

/* Driver program to test above function */
int main()
{
  int arr[] = {1, 101, 2, 3, 100, 4, 5};
  int n = sizeof(arr)/sizeof(arr[0]);
  printf("Sum of maximum sum increasing subsequence is %d\n",
         maxSumIS( arr, n ) );

  getchar();
  return 0;
}
```

Time Complexity: O(n^2)