Maximum sum increasing subsequence

Given an array **arr** of **N** positive integers, the task is to find the **maximum sum increasing subsequence** of the given array.

Example 1:

```
Input: N = 5, arr[] = {1, 101, 2, 3, 100}
Output: 106
Explanation:The maximum sum of a
increasing sequence is obtained from
{1, 2, 3, 100}
```

Example 2:

```
Input: N = 3, arr[] = {1, 2, 3}
Output: 6
Explanation:The maximum sum of a
increasing sequence is obtained from
{1, 2, 3}
```

Your Task:
You don't need to read input or print anything. Complete the function `maxSumIS()` which takes **N** and array **arr** as input parameters and returns the maximum value.

Expected Time Complexity: O(**N**^{2})
Expected Auxiliary Space: O(**N**)

Constraints:

$1 \leq N \leq 10^3$

$1 \leq arr[i] \leq 10^5$

```python
class Solution:
    def maxSumIS(self, Arr, n):
        # code here
        dp = [0]*n
        dp[0] = Arr[0]
        for i in range(1,n):
            temp = 0
            for j in range(i):
                if Arr[j]<Arr[i]:
```

```
                    temp = max(temp,dp[j])
            temp = temp+Arr[i]
            dp[i] = temp
        return max(dp)
```