**Box Stacking :-**

You are given a set of **N** types of rectangular 3-D boxes, where the ith box has height **h**, width **w** and length **l**. You task is to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the **dimensions** of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base.It is also allowable to use multiple instances of the same type of box. You task is to complete the function **maxHeight** which returns the height of the highest possible stack so formed.

NOTE:- You can twerk the box in any way such that any side can become length or width or height,and you can use any number of instances of the same box.

**Note:**  
Base of the lower box should be strictly larger than that of the new box we're going to place. This is in terms of both length and width, not just in terms of area. So, two boxes with same base cannot be placed one over the other. So base rectangle dimensions of lower box should be greater than box above it.

**Example 1:**

**Input:**

n = 4

height[] = {4,1,4,10}

width[] = {6,2,5,12}

length[] = {7,3,6,32}

**Output:** 60

**Explanation:** One way of placing the boxes is

as follows in the bottom to top manner:

(Denoting the boxes in (l, w, **h**) manner)

(12, 32, **10**) (10, 12, **32**) (6, 7, **4**) (5, 6, **4**)

(4, 5, **6**) (2, 3, **1**) (1, 2, **3**)

Hence, the total height of this stack is

10 + 32 + 4 + 4 + 6 + 1 + 3 = 60.

No other combination of boxes produces a height

greater than this.

â€‹**Example 2:**

**Input**:

n = 3

height[] = {1,4,3}

width[] = {2,5,4}

length[] = {3,6,1}

**Output:** 15

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **maxHeight()** which takes three arrays height[], width[], length[], and N as a number of boxes and returns the **highest possible stack height** which could be formed.

**Expected Time Complexity** : O(N\*N)  
**Expected Auxiliary Space**: O(N)

**Constraints:**  
1<=N<=100  
1<=l,w,h<=100