CS575 Fall 2019

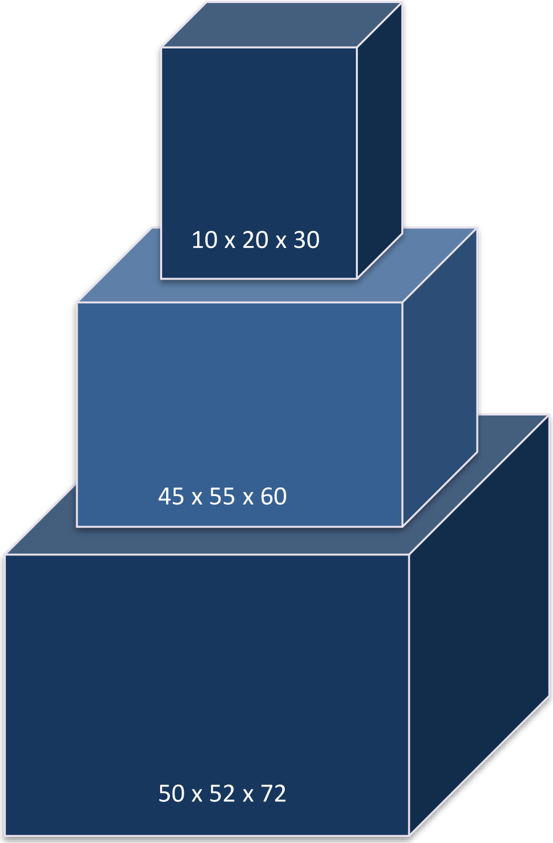
Box Stacking Problem

Chaitanya Kulkarni

Problem Statement

- Given n types of rectangular 3D boxes, with length (l), breadth (b) and height(h)
- Goal is to stack them on each other such that height of stack is MAXIMUM
- Constraints –

  The stack should be such that any box should have its base dimensions (length & breadth) strictly smaller than box below.
- It is allowed to use the multiple instances of the box and can rotate the box in any direction



Algorithm Description

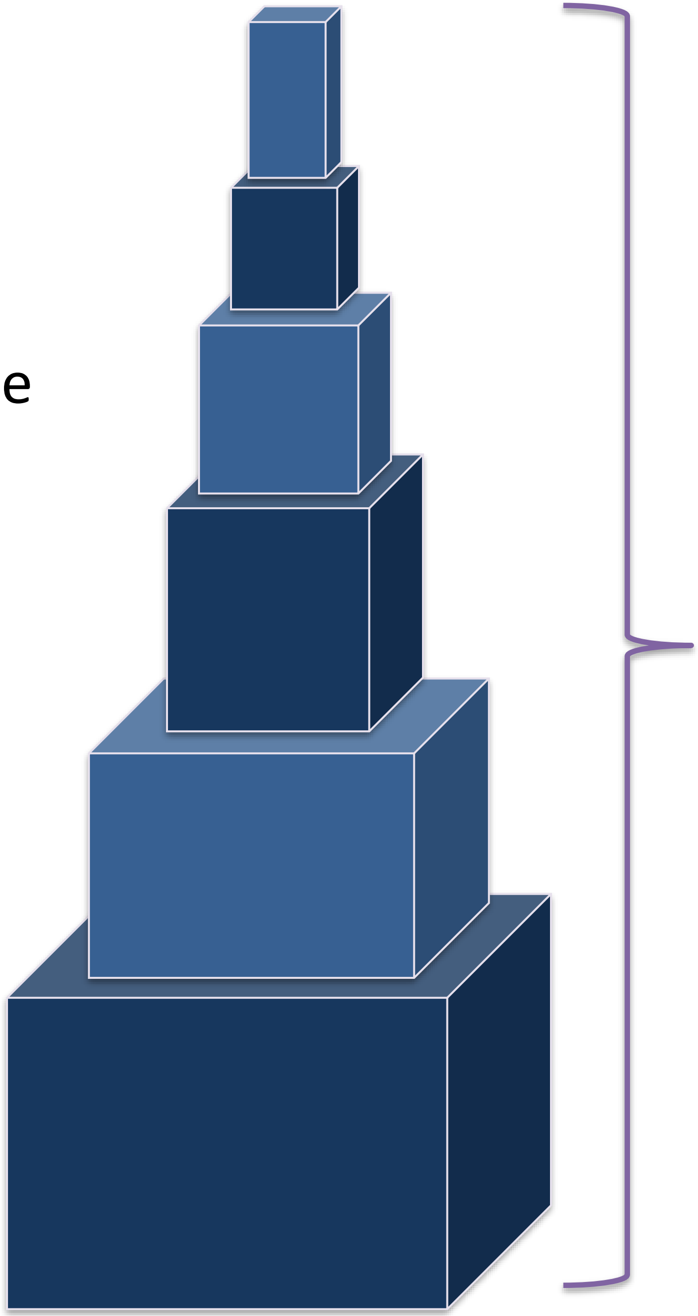
This is the variant of Longest Increasing Subsequence (LIS)

1. Generate all 3 rotations of all boxes. Hence, the size of rotation array becomes 3 times the size of original array.
2. Sort the above generated 3n boxes in decreasing order of base area.
3. After sorting the boxes, the problem is same as LIS with following optimal substructure property.
   MaxSH(i) = { Max (MaxSH(j)) + height(i) }
   where j < i and length(j) > length(i) and breadth(j) > breadth(i).
   If there is no such j then MaxSH(i) = height(i)
4. To get overall maximum height, we return max(MaxSH(i))
   where 0 < i < n

Results (if any)

- The expected result should be the Maximum height of the stack
- Time Complexity

  $T(n) = O(n^2)$
- Space Complexity = $O(n)$



Max height = 162