November 4, 2015

CS 577 - Homework 7
Sejal Chauhan, Vinothkumar Siddharth, Mihir Shete

1 Graded written problem

Input: Given a sheet AXB, there are (a_i, b_i), where i is a positive integer, sculptures that can be formed.

Output: Maximize the number of sculptures that can be made from the starting sheet of paper while minimizing frayed edges that must be visible

1.1 Algorithm

Basically, we are using a top-down Dynamic programming approach to solve this problem. The algorithm in pseudocode looks as follows:

Algorithm 1 Maximize the number of sculptures and minimize frayed edges that must be visible.

procedure FIND-MAX-SCULPTURES(A, B, $S[1..n]$)
  $MaxSculpture[A][B] \leftarrow 0$
  $FrEdges[A][B] \leftarrow 0$
  for `i = 1..A` **do**
    for `j = 1..B` **do**
      for `s = S[1]..S[n]` **do**
        if s_{length}<i AND $s_{breadth}$<j **then** ▷ checking if the length and breadth are less than A and B
          $Max1 \leftarrow 1 + MaxSculpture[i - s_{length}][s_{breadth}] + MaxSculpture[i][j - s_{breadth}]$
          $FrEdges1 \leftarrow CalFrayedEdges()$
          $Max2 \leftarrow 1 + MaxSculpture[i - s_{length}][j] + MaxSculpture[s_{length}][j - s_{breadth}]$
          $FrEdges2 \leftarrow CalFrayedEdges()$
        if s_{length}<B 1 AND $s_{breadth}$<A 2 **then**
          $Max3 \leftarrow 1 + MaxSculpture[s_{breadth}][j - s_{length}] + MaxSculpture[i - s_{breadth}][j]$
          $FrEdges3 \leftarrow CalFrayedEdges()$
          $Max4 \leftarrow 1 + MaxSculpture[i - s_{breadth}][s_{length}] + MaxSculpture[i][j - s_{length}]$
          $FrEdges4 \leftarrow CalFrayedEdges()$
        $MaxSculpture[i][j] \leftarrow \max Max1, Max2, Max3, Max4$
        $FrEdge[i][j] \leftarrow \max FrEdges1, FrEdges2, FrEdges3, FrEdges4$
  return $\max(MaxSculpture[][])$

As we can see, we are using 3 loops in our procedure, first one is used to iterate over the edge with length A. The second one will go over all the permutations of length B. While considering each permutation we also check if the frayed edges of the total sculptures that are made. So the performance of our algorithm will be $O(n^3)$.