Design and Analysis of Algorithm

Seminar Report

Group no.- 09

Minimum number of squares that can be cut from a given rectangle

Sardar Vallabhbhai National Institute of Technology, Surat

Date of submission – 18/04/2021

Abstract- For a Rectangular shape, many squares can be cut of integer side length. For e.g., maximum m x n squares can be cut of side length 1 from a m x n rectangle, since there are many combinations possible for choosing the squares from a rectangle and from that we have to select minimum number of squares, it is not an easy task to do, so we have to find the algorithm, that can help us to find the minimum number of squares that can be cut from a rectangle. This problem is just a special case of a group of problems studied by mathematicians.

I. INTRODUCTION: PROBLEM STATEMENT

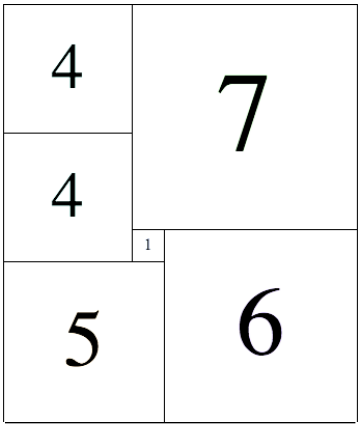
The computational problem is given as "Find the minimum number of squares that can be cut from a given rectangle having side length m and n. The side length of the squares and given rectangle should be positive integers only.". That means, the inputs we will take as two integer value and these values will denote the side length of rectangle and the expected output is an integer which is the total count of minimum number of squares that can be cut from the rectangle. As we know that there are so many combinations possible to select or cut squares from a rectangle. As an example, for a 2 x 3 rectangle, we can get six 1 x 1 squares or we can get one 2 x 2 square and two 1 x 1 squares. But here we have to second choice since it is giving minimum no. of squares (3). This task looks easy for smaller side length inputs m and n, but to find the minimum number of squares from a rectangle, which has much bigger side length compared to this example and the reason is, since we have greater side length so there will be many possible ways to cut the squares from it and we have to choose, for which choice, we will get least number of squares.

Examples -

Input : m = 11 and n = 13

Output : 6

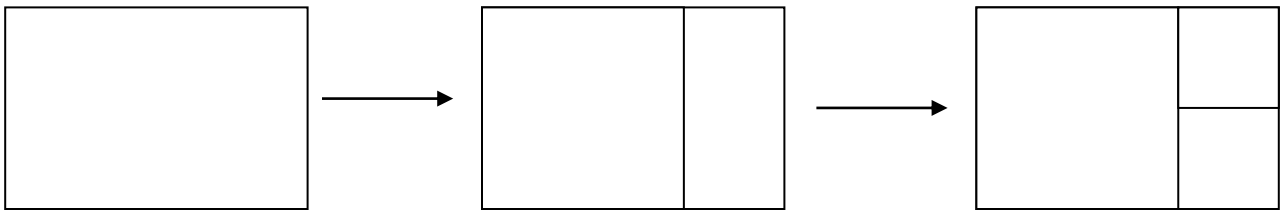
Explanation : 1 (7 x 7) square +
1 (6 x 6) square +
1 (5 x 5) square +
2 (4 x 4) squares +
1 (7 x 7) square
= 6 squares



II. ALGORITHM DESCRIPTION

For solving this Computational problem, we have tried to implement various algorithmic approach, such as Greedy approach, Recursive approach, Dynamic Programming approach and finally we have used Backtracking with Dynamic Programming. In Greedy and Dynamic Programming both, there are many exceptional cases, for which output is not right so we have used Backtracking which is using the result of Dynamic Programming to check all the possibility in efficient manner. Brief description for all the approaches, that are mentioned above, is following –

Greedy Solution – The greedy choice for this problem is, to cut the maximum side length square possible in the given rectangle. After this repeat the same step for the remaining rectangular part until we don't get a perfect square as the remaining part after cutting.



Here, we can say that the maximum side length square will have the same length of the minimum of m and n, where m and n are the side length of rectangle. Every time, when we cut square from the rectangle, the side length of the rectangle updates and then we consider this remaining part as the rectangle and repeat the same process again and again.

This algorithm is not right for every input, especially for greater value of m and n. For smaller values of m and n also, we have many exceptions. These cases, we can reduce using Dynamic Programming approach.

Dynamic Programming Solution – We try to write a function called minimumSquare, which tries to split the rectangle at some position. The function is called recursively for both parts. Try all possible splits and take the one with minimum result. The base case is when both sides are equal i.e. the input is already a square, in which case the result is We are trying to find the cut which is nearest to the center which will lead us to our minimum result.

Assuming we have a rectangle with width is N and height is M.

- if (N == M), so it is a square and nothing need to be done.
- Otherwise, we can divide the rectangle into two other smaller one (N – x, M) and (x, M), so it can be solved recursively.
- Similarly, we can also divide it into (N, M – x) and (N, x)

Also, we need to be aware of some edge case here which can lead to the wrong output. We can put already calculated right output in the minimumSquare function for these exceptional cases.

Backtracking with Dynamic Programming – We use DFS / backtracking, in which we start stacking squares from any of the four corners of the given rectangle. Then we try largest possible squares first to prune DFS and we use results generated by (naive) DP discussed above as smaller upper bounds of true answers. By using this we check for all the possibilities and try to solve the issues, which we were facing in naïve dynamic programming approach.

III. Algorithm Analysis

Approach	Time Complexity	Space Complexity
Greedy Solution	O(n)	O(1)
Dynamic Programming	O(mn * max(m, n))	O(mn)
Backtracking with DP	O(m^n)	O(n)

IV. Hardware / Software Description

- **System type:** 64-bit OS, x64-based processor
- **OS:** Windows 10 (Version 10.0.18363 Build 18363)
- **Processor:** Intel Core i5-8300H CPU @2.30GHz
  4 Core(s), 8 Logical Processor(s)
- **Compiler:** GCC 8.1.0

V. Conclusion

After solving this Computational problem, we can say that this problem is an interesting problem and it is hard to write algorithm for this question. Some approaches, we tried but those approaches are not good enough to get the right output for every type of inputs and especially for the greater value of m and n. So, we tried Backtracking to checking all the possibility and this approach is costly in terms of time complexity but this approach is right for all the input. This problem will help us to solve some more Computational problems in Mathematics.

Authors

- **Shubham kumar jayswal** – Enrollment no.-U19CS070
- **Akkaladevi Shreyas** – Enrollment no.-U19CS069
- **Chakka Vasudev** – Enrollment no.-U19CS017
- **Shaik R Irshad Areez Ahmmad** – Enrollment no.-U19CS094