**Project - 2 Report**

**UTA ID:**

1002144180: Pranavakesava Reddy Madduri

1002175121: Lokeshwar Kodipunjula

**Task 1: Cutting Rods for Maximum Revenue**

Time Complexity: O(N^2)

**Input:**

We read the input1.txt file. Each problem is given by two consecutive lines, where the first line contains the length of the rod (n), and the second line contains an array of prices (p). We extract two arrays from the input file, one for the rod length and another one for the corresponding prices.

**Implementation:**

We implemented the Cut-Rod algorithm to find the maximum revenue for cutting a rod of length into pieces with different prices. The algorithm recursively goes through all possible combinations to find the optimal cost to cut the rod.

**Experimental Results:**

**Test Case 1:**

**Input 1:** n = 3, p = [2, 3, 4]

**Test Case 2:**

**Input 2:** n = 5, p = [1, 2, 4, 4, 5]



**Explanation:**

The Cut-Rod algorithm does this by considering all possible cuts and calculating the maximum revenue for each subproblem. It builds a table to store optimal solutions to subproblems, ensuring an efficient and optimal solution overall.

**Task 2: Wood Board Cutting**

**Input:**

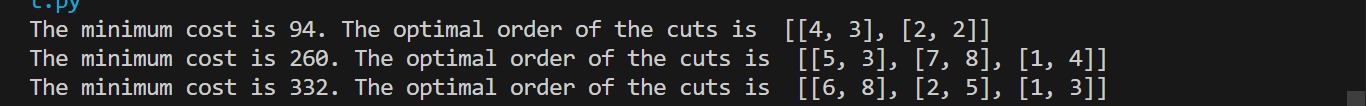
We read the input2.txt file, where the first line contains the width and height of the wood board, and the following are the positions of cuts or sets of cuts. The cut points from the file are processed to store the two sets of dimensions into two separate 2D arrays. We applied the Wood Board Cutting algorithm, and the minimum total cost along with the optimal order of cuts are figured out.

**Implementation:**

Wood Board Cutting algorithm determines the minimum total cost of cutting a wood board with specified dimensions (‘width’ \* ‘height’), which involves dividing the board into four pieces with each cut, creating four smaller sub-pieces. It does all permutations and combinations of the board-cutting with the given cuts or points and comes up with minimum cost and optimal order of the cuts.

**Experimental Results:**

**Test Cases:**



**Explanation:**

Here we used dynamic programming which is the Wood Board Cutting algorithm to find the minimum total cost of making cuts in a particular order. It considers the costs associated with each cut and optimally arranges them to minimize the overall cost. It looks at how much each cut costs and organizes them in the best way to spend the least amount overall.

**References:**

1. Lecture PPTS
2. <https://www.geeksforgeeks.org/cutting-a-rod-dp-13/>
3. <https://www.geeksforgeeks.org/matrix-chain-multiplication-dp-8/>
4. <https://www.youtube.com/watch?v=prx1psByp7U>

**Honor Code**

