|  |
| --- |
| **Ques – 7 DSA LAB ACTIVITY ASSESSMENT**  **21CSC201J – Data Structures & Algorithms**  *Submitted by*  **Arnav Anand [Reg. No.: RA2311026010259]**  **Devansh Gupta [Reg. No.: RA2311026010260]**  **B.Tech. CSE – AI & ML**  **SRMIST-01.jpg**  **SCHOOL OF COMPUTING**  **COLLEGE OF ENGINEERING AND TECHNOLOGY**  **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  **(Under Section 3 of UGC Act, 1956)**  S.R.M. NAGAR, KATTANKULATHUR – 603 203  CHENGALPATTU DISTRICT  **October 2024** |

**Problem Statement**

Question 7:

Given a rod of length n and list of prices of rod of length i where 1<=i<=n find the optimal way to cut rod into smaller rods in order to maximize profit.

Input Format

To get the rod length.

Output Format

Outputs the profit.

Constraints Integers only.

Sample Input :

4

Sample Output :

10

**Algorithm**

1. **Rod Cutting Function (Rod):**
   * Input: rate[] (an array of prices for different lengths of the rod), n (the current length of the rod).
   * Base Case: If the rod length is 0, return 0 because no rod means no revenue.
   * Recursive Step: For each possible length i (from 1 to n), compute the revenue by cutting the rod into a piece of length i (with price rate[i-1]) and recursively finding the maximum revenue for the remaining rod length (n-i).
2. **Recursive Approach:**
   * For every possible cut length i, it tries to calculate the maximum revenue recursively for the remaining part of the rod (n-i).
   * The recursion continues until it reduces the rod length to zero.
3. **Main Function:**
   * Takes user input for the size of the rod.
   * Calls the Rod function with the price[] array and the rod size.

**Example Walkthrough:**

Given price[] = {1, 5, 8, 9, 10, 17, 17, 20} and rod size 4:

* It will check possible cuts like:
  + Cut the rod into lengths [1, 3]
  + Cut the rod into lengths [2, 2]
  + Cut the rod into lengths [3, 1]
  + Cut the rod into lengths [4, 0]
* The function recursively computes the maximum profit from each combination.

**Program Code**

#include <iostream>

using namespace std;

// Function which returns max profit

int Rod(int rate[], int n)

{

    // Base case

    if (n == 0)

    {

       return 0;

    }

//  Variable for storing the result

    int res = 0;

    // THis loop will check for the all possible cases

    for (int i=0; i<n; i++)

    {

        // This will recursively calculate tyhe all possible solutions and give the finbal output

       res = max(res, rate[i] + Rod(rate, n-i-1));

    }

    // Returning the result

    return res;

}

int main()

{

// Price Array

    int price[] = {1, 5, 8, 9, 10, 17, 17, 20};

    // Taking the size of the rod

    int size;

    cout<<"Enter the Size"<<endl;

    cin>>size;

    // Computing max profit

    cout<<"Max Profit is"<<endl;

    cout << Rod(price, size)<<endl;

    return 0;

}

**Sample Input and Output**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**GitHub Link**

<https://github.com/Red-Spidy/DSA>

**Result-**

**Output Verification :-**

The above code was compiled and executed successfully no problems were found.

**Performance :-**

The above code follows exponential time complexity O(2^n).