Project 3: Dynamic-Programming Report

Phi Stanton

CSCI 406: Algorithms

## Project Description:

Our goal for this project was to maximize the length of wood we would get from a sawmill by picking which end of the tree to cut at each turn, this is in direct competition with someone else implementing a proper algorithm.

The recurrence relation for this problem was as given:

Text, Word

Description automatically generated

# Recursive Algorithm:

For this step, I was tasked to create a recursive algorithm using the given recurrence relation, and run it on the array [33, 28, 35, 25, 29, 34, 28, 32].

1. Including the initial call, how many calls are made to the function?

Including the initial call, and all redundant calls, there are a total of 85 calls made to the function.

1. Derive the asymptotic complexity and provide an exact bound for the recursive algorithm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N (Resolution of 316ns)** | Trial 1 (Seconds) | Trial 2 (Seconds) | Trial 3 (Seconds) | Average |
| 15 | 0.005980 | 0.005946 | 0.005984 | 0.005967 |
| 20 | 0.119211 | 0.119124 | 0.122677 | 0.120337 |
| 23 | 1.444597 | 1.505584 | 1.532705 | 1.494295 |
| 25 | 5.988732 | 5.968992 | 5.829551 | 5.929092 |
| 26 | 7.506632 | 7.956427 | 7.762956 | 7.742005 |

# Dynamic Programming Algorithm:

1. Implement this function:

Code found in Appendix A

1. Derive the asymptotic complexity and provide an exact bound for the dynamic programming algorithm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N (Resolution of 316ns)** | Trial 1 (Seconds) | Trial 2 (Seconds) | Trial 3 (Seconds) | Average |
| 50 | 0.005980 | 0.005946 | 0.005984 | 0.005967 |
| 100 | 0.119211 | 0.119124 | 0.122677 | 0.120337 |
| 200 | 1.444597 | 1.505584 | 1.532705 | 1.494295 |
| 400 | 5.988732 | 5.968992 | 5.829551 | 5.929092 |
| 800 | 7.506632 | 7.956427 | 7.762956 | 7.742005 |