

CO322: Data Structures and Algorithms

Lab 03: HR problem set 03

WIJERATHNE E.S.G

E/18/397

01. Stone Division

We cannot do piling if $n = 0$ or $n = 1$. So, this will be our base case, and we will return 0. Then we will have to check if we have already performed the task. For that, we have to store pile values for particular n values. If we have a match, we can return that pile count.

In the main function, we go through the set of numbers in S and look for the combinations using the recursive method.

02. Simplified Chess Engine

First, define paths for chess pieces for each piece on the board, then try for each combination until we have won or lost.

03. Crossword puzzle

We first require a method to locate all of the accessible zones. The vertical direction makes this a little challenging. To ascertain whether the zone fits the keyword, a function is required. We'll go backward. So, the functions `forward()` and `reverse()` are required. Prior to recursion, `forward()` places the keyword in the appropriate location; the following recursion, `reverse()` removes it. Finally, we require the solution recursive function `()`. The wedding portion is now. It will attempt to fill each zone with text in a recursive fashion by finding the longest possible spaces. It will perform a rollback if it fails.