CS250 Winter 2003

Week 3, Lecture 2: Recursion
Recursive matrix multiplication

Follows a recursive algorithm for the product of two arbitrary square matrices $A$ and $B$.

Lines 10-20 use Strassen’s method, and we assume that more basic matrix operations (like assignment and addition) are already implemented.
Prod(A, B) {
    if n = 1 then
        C := A x B // integer multiplication
    else {
        parity := n mod 2
        if parity = 1 then {
            add a row and a column of 0's to A and B
            n := n + 1
        }
        let A_11 A_12 := A and B_11 B_12 := B
        A_21 A_22
        B_21 B_22
        M_1 := Prod(A_12 - A_22, B_21 + B22)
        M_2 := Prod(A_11 + A_12, B_11 + B22)
        M_3 := Prod(A_11 - A_21, B_11 + B12)
        M_4 := Prod(A_11 + A_12, B_22)
        M_5 := Prod(A_11, B_12 - B_22)
        M_6 := Prod(A_22, B_21 - B_11)
        M_7 := Prod(A_21 + A_22, B_11)
        C_11 := M_1 + M_2 - M_4 + M_6
        C_12 := M_4 + M_5
        C_21 := M_6 + M_7
        C_22 := M_2 - M_3 + M_5 - M_7
        let C_11 C_12 := C
        C_21 C_22
        if parity = 1 then {
            remove the last row and the last column from C
            n := n - 1
        }
    }
    return C
}