Ackerman function

The Ackermann function is a classic example of a recursive function, notable especially because it is not a primitive recursive function. It grows very quickly in value, as does the size of its call tree.

The Ackermann function is usually defined as follows:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1,A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

Its arguments are never negative and it always terminates.

Write a function which returns the value of $A(m,n)$. Arbitrary precision is preferred (since the function grows so quickly), but not required.

⚗	`ack` should be a function.
⚗	`ack(0,0)` should return 1.
⚗	`ack(1,1)` should return 13.
⚗	`ack(2,5)` should return 13.
⚗	`ack(3,3)` should return 61.