Ackermann-Funktion

$$
\begin{aligned}
f(0,n) &= n+1 \\
f(m,0) &= f(m-1,1) \\
f(m,n) &= f(m-1,f(m,n-1))
\end{aligned}
$$

```
def ack(m, n):
    if m == 0:
        return n + 1
    elif n == 0:
        return ack(m - 1, 1)
    else:
        return ack(m - 1, ack(m, n - 1))
```

$A(m,n)$	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
$m=0$	1	2	3	4	5	6
$m=1$	2	3	4	5	6	7
$m=2$	3	5	7	9	11	13
$m=3$	5	13	29	61	125	253
$m=4$	13	65533	$2^{65536}-3$	$2^{2^{65536}}-3$	$2^{2^{2^{65536}}}-3$	$2^{2^{2^{2^{65536}}}}-3$
$m=5$	65533					
$m=6$	$2^{2^{2^{\cdots}}}$ (65536mal) -3					

$A(1,n) = n+2$
$A(2,n) = 2n+3$
$A(3,n) = 2^{n+3}-3$

$A(4,2)$ hat schon 19729 Stellen.
Die Zahl ist größer als die (vermutete) Anzahl der Atome im Weltraum.

Genau die primitiv-rekursiven Funktionen können mit Additionen, Zuweisungen und loop-Schleifen programmiert werden. Für die Ackermann-Funktion (1928) trifft das nicht zu.
Sie wächst schneller als jede primitiv-rekursive Funktion.

Ackermann-Funktion anschaulich

$$
\begin{aligned}
f(0,n) &= n+1 \\
f(m,0) &= f(m-1,1) \\
f(m,n) &= f(m-1,f(m,n-1))
\end{aligned}
$$