

SOURCE CODE 1. Unranking: Funktionen von beschränktem Wachstum aus dem Rang (in lexikographischer Ordnung).

```
def unrank_function_of_restricted_growth(rank, nnn):  
    """Berechne die Funktion von beschränktem Wachstum, die in  
lexikographischer Ordnung den gegebene Rang hat."""  
    # Vorbereitung: r-Bell-Zahlen und Funktion, die sie "ausliest".  
    the_table = compute_r_Bell_numbers(nnn)  
    def local_d(i,m):  
        return the_table[i-1][m-1]  
  
    # Vorbereitung der Arrays:  
    the_func = np.zeros(nnn, dtype=int)  
    the_maxima = np.zeros(nnn, dtype=int)  
    # Die Funktion hat  $f(1) = 1$ , und das  $\max(\{f(1)\})$  ist ebenso 1:  
    the_func[0] = the_maxima[0] = 1  
  
    # Berechne die Funktion aus dem Rang:  
    for i in range(1,nnn):  
        m = the_maxima[i-1]  
        # Achtung: Indices in Python beginnen bei 0  
        d_im = local_d(i+1,m)  
        # Schauen wir mal, ob the_func[i] = m+1 möglich ist:  
        m_d_im = m*d_im  
        if m_d_im <= rank:  
            # Wenn ja: Setze the_func[i] und neues Maximum:  
            the_func[i] = the_maxima[i] = m+1  
            rank -= m_d_im  
        else:  
            # Wenn nein: Berechne the_func[i] durch Division mit Rest ...  
            the_func[i] = rank//d_im + 1  
            rank %= d_im  
            # ... und das alte Maximum wird "weiterverwendet":  
            the_maxima[i] = m  
  
    return the_func
```

Umkehrung der Rangfunktion:  $\text{rank} \mapsto \text{function}f$ . Wir brauchen hier als Vorbereitung eine Matrix von Zahlen, aus der wir dann Werte ablesen, die wir in der Berechnung verwenden.