

Lösningar

Lösning till Talföljder

0, 1, 2, 3, 6, 11, 20

För att beräkna ett tal i följderna behöver man bara känna till de tre närmast föregående talen. Beräkningsordningen blir stigande index från basfallen och uppåt.

```
Talfoljd(n) =  
  s[0] = 0  
  s[1] = 1  
  s[2] = 2  
  for i = 3 to n do  
    s[i] = s[i-1]+s[i-2]+s[i-3]  
  return s[n]
```

□

Lösning till Generaliserade talföljder

Vi måste börja med ett basfall, men om vi tänker oss en matris C som ska fyllas i med rätt värden, så har hela översta raden (där $i = 0$) och hela vänstra kolumnen (där $j = 0$) värdet 0, som inte beräknas rekursivt. En typisk beräkningsordning för ett program skulle vara en rad i taget uppifrån eller en kolumn i taget från vänster till höger. Vi väljer rad för rad, och sätter värdena på första raden. Därefter loopar vi över alla andra rader och sätter värdena kolumn för kolumn. Beräkningsordningen fungerar, eftersom den information vi behöver ha för att beräkna ett nytt värde utom basfallen bara är vad som stått på tidigare rader, samt indexet för den rad vi befinner oss på. Vi kommer att kunna beräkna hela C . Här är vår algoritm:

```
for  $j \leftarrow 0$  to  $n$ 
     $C[0, j] \leftarrow 0$ 
for  $i \leftarrow 1$  to  $m$ 
     $C[i, 0] \leftarrow 0$ 
    for  $j \leftarrow 1$  to  $n$ 
         $C[i, j] \leftarrow C[i - 1, j] + C[i - 1, j - 1] + 1$ 
```

Tiden domineras av den nästlade for-slingan och är alltså $\Theta(nm)$.

□

Lösning till Matriskedjemultiplikation

Basfallen är $M[i, i]$, det vill säga diagonalen i (den triangulära) matrisen M . Beräkningsordning: Beräkna elementen efter stigande avstånd från diagonalen, alltså så att indexen i och j skiljer sig mer och mer.

```
OptimalMatmul( $M[1..n, 1..n]$ ,  $r$ ) =  
  for  $i \leftarrow 1$  to  $n$  do  $M[i, i] \leftarrow 0$   
  for  $len \leftarrow 1$  to  $n - 1$   
    for  $i \leftarrow 1$  to  $n - len$   
       $j \leftarrow i + len$   
       $M[i, j] \leftarrow \infty$   
      for  $k \leftarrow i$  to  $j - 1$   
         $t \leftarrow M[i, k] + M[k + 1, j] + r_{i-1} \cdot r_k \cdot r_j$   
        if  $t < M[i, j]$  then  $M[i, j] \leftarrow t$ 
```

Det minimala antalet multiplikationer beräknas av `OptimalMatmul( $M$ ,  $r$ )`. Svaret finns då i $M[1, n]$. Algoritmen har komplexiteten $O(n^3)$. □
