Semesterarbeit Teil 1a

Aufgabe 1 und 2

`fibonacci_count.py`

Das Programm erwartet die Eingabe eines Integers `max_fibonacci` und berechnet dann die Fibonacci Zahlen von 0 bis zu dieser Obergrenze.

Für Aufgabe 2 wird die Anzahl Aufrufe der rekursiven Funktion `fib(n)` ausgegeben, welche zur Berechnung nötig war. Diese Anzahl wird über die globale Variable `total_calls` gezählt.

```
Fibonacci numbers for range 0 to 10
n:   0    fib(n):   0    total_calls:    1
n:   1    fib(n):   1    total_calls:    1
n:   2    fib(n):   1    total_calls:    3
n:   3    fib(n):   2    total_calls:    5
n:   4    fib(n):   3    total_calls:    9
n:   5    fib(n):   5    total_calls:   15
n:   6    fib(n):   8    total_calls:   25
n:   7    fib(n):  13    total_calls:   41
n:   8    fib(n):  21    total_calls:   67
n:   9    fib(n):  34    total_calls:  109
n:  10    fib(n):  55    total_calls:  177
```

Aufgabe 3

Die Anzahl Funktionsaufrufe ergibt diese Folge:

```
    1, 3, 5, 9, 15, 25, 41, 67, 109, 177, ...
```

Gegenüberstellung Anzahl Funktionsaufrufe mit Fibonacci Zahlen:

```
    Funktionsaufrufe:   1   1   3   5   9  15  25  41  67 109 177
    Fibonacci Zahl:     0   1   1   2   3   5   8  13  21  34  55
```

Vermutung für die Anzahl Funktionsaufrufe `total_calls` zur Berechnung der n-ten Fibonacci Zahl:

```
total_calls = 2 * fib(n + 1) - 1
```

Benötigt also selbst wieder die Fibonacci Funktion.

Aufgabe 4

`fibonacci_time.py`

`fibonacci_module.py`

Funktion `fib(n)` jetzt in separatem Modul untergebracht. Gleich wie erste Implementierung, aber ohne globalen Zähler (wegen Einfluss auf Zeitmessung).

Das Programm `fibonacci_time.py` ruft `fib(n)` auf und gibt die gemessenen Zeiten aus. Das sieht nach exponentiellem Wachstum für die "elapsed seconds" aus.

```
Fibonacci numbers for range 0 to 40
n:   0    fib(n):          0    elapsed seconds: 0.0
n:   1    fib(n):          1    elapsed seconds: 0.0
n:   2    fib(n):          1    elapsed seconds: 0.0
n:   3    fib(n):          2    elapsed seconds: 0.0
n:   4    fib(n):          3    elapsed seconds: 0.0
n:   5    fib(n):          5    elapsed seconds: 0.0
n:   6    fib(n):          8    elapsed seconds: 0.0
n:   7    fib(n):         13    elapsed seconds: 0.0
n:   8    fib(n):         21    elapsed seconds: 0.0
n:   9    fib(n):         34    elapsed seconds: 0.0
n:  10    fib(n):         55    elapsed seconds: 0.0
n:  11    fib(n):         89    elapsed seconds: 0.0
n:  12    fib(n):        144    elapsed seconds: 0.0
n:  13    fib(n):        233    elapsed seconds: 0.0009992122650146484
n:  14    fib(n):        377    elapsed seconds: 0.0
n:  15    fib(n):        610    elapsed seconds: 0.0
n:  16    fib(n):        987    elapsed seconds: 0.0010001659393310547
n:  17    fib(n):       1597    elapsed seconds: 0.0010001659393310547
n:  18    fib(n):       2584    elapsed seconds: 0.0019998550415039062
n:  19    fib(n):       4181    elapsed seconds: 0.003000020980834961
n:  20    fib(n):       6765    elapsed seconds: 0.004999399185180664
n:  21    fib(n):      10946    elapsed seconds: 0.007999897003173828
n:  22    fib(n):      17711    elapsed seconds: 0.011999845504760742
n:  23    fib(n):      28657    elapsed seconds: 0.01999950408935547
n:  24    fib(n):      46368    elapsed seconds: 0.031000614166259766
n:  25    fib(n):      75025    elapsed seconds: 0.04999876022338867
n:  26    fib(n):     121393    elapsed seconds: 0.08199858665466309
n:  27    fib(n):     196418    elapsed seconds: 0.1224210262298584
n:  28    fib(n):     317811    elapsed seconds: 0.23323798179626465
n:  29    fib(n):     514229    elapsed seconds: 0.3434438705444336
n:  30    fib(n):     832040    elapsed seconds: 0.5441131591796875
n:  31    fib(n):    1346269    elapsed seconds: 0.9086081981658936
n:  32    fib(n):    2178309    elapsed seconds: 1.4448347091674805
n:  33    fib(n):    3524578    elapsed seconds: 2.365004062652588
n:  34    fib(n):    5702887    elapsed seconds: 3.8220174312591553
n:  35    fib(n):    9227465    elapsed seconds: 6.232163667678833
n:  36    fib(n):   14930352    elapsed seconds: 10.18271780014038
n:  37    fib(n):   24157817    elapsed seconds: 16.28048849105835
n:  38    fib(n):   39088169    elapsed seconds: 26.166386365890503
```

```
n:  39    fib(n):   63245986    elapsed seconds: 42.51878809928894
n:  40    fib(n):  102334155    elapsed seconds: 69.38479280471802
```

Aufgabe 5

`fibonacci_caching.py`

`fibonacci_module.py`

Da in der rekursiven Implementierung die kleineren Fibonacci Zahlen für jede Verästelung immer wieder neu berechnet werden müssen, hatte ich die Idee, alle bereits berechneten Fibonacci Zahlen in einem Cache zu speichern. Der Code für die neue Funktion `fib_caching` ist im Modul `fibonacci_module.py`:

```
def fib_caching(n, cache={}):
    """This implementation for calculating a Fibonacci number
    caches the already calculated numbers.
    """
    cached_fibonacci_number = cache.get(n, None)
    if cached_fibonacci_number is not None:
        return cached_fibonacci_number
    if n == 0 or n == 1:
        cache[n] = n
        return n
    fibonacci_number = fib_caching(n - 1, cache) + fib_caching(n - 2, cache)
    cache[n] = fibonacci_number
    return fibonacci_number
```

Ausprobiert mit `fibonacci_caching.py` zeigt sich, dass die Performance massiv besser ist, nicht einmal mehr im Sekundenbereich messbar:

```
Fibonacci numbers for range 0 to 100
n:   0    fib(n):                    0    elapsed seconds: 0.0
n:   1    fib(n):                    1    elapsed seconds: 0.0
n:   2    fib(n):                    1    elapsed seconds: 0.0
n:   3    fib(n):                    2    elapsed seconds: 0.0
n:   4    fib(n):                    3    elapsed seconds: 0.0
n:   5    fib(n):                    5    elapsed seconds: 0.0
n:   6    fib(n):                    8    elapsed seconds: 0.0
n:   7    fib(n):                   13    elapsed seconds: 0.0
n:   8    fib(n):                   21    elapsed seconds: 0.0
n:   9    fib(n):                   34    elapsed seconds: 0.0
n:  10    fib(n):                   55    elapsed seconds: 0.0
n:  11    fib(n):                   89    elapsed seconds: 0.0
n:  12    fib(n):                  144    elapsed seconds: 0.0
n:  13    fib(n):                  233    elapsed seconds: 0.0
n:  14    fib(n):                  377    elapsed seconds: 0.0
n:  15    fib(n):                  610    elapsed seconds: 0.0
n:  16    fib(n):                  987    elapsed seconds: 0.0
n:  17    fib(n):                 1597    elapsed seconds: 0.0
```

```
n:  18    fib(n):                2584    elapsed seconds: 0.0
n:  19    fib(n):                4181    elapsed seconds: 0.0
n:  20    fib(n):                6765    elapsed seconds: 0.0
n:  21    fib(n):               10946    elapsed seconds: 0.0
n:  22    fib(n):               17711    elapsed seconds: 0.0
n:  23    fib(n):               28657    elapsed seconds: 0.0
n:  24    fib(n):               46368    elapsed seconds: 0.0
n:  25    fib(n):               75025    elapsed seconds: 0.0
n:  26    fib(n):              121393    elapsed seconds: 0.0
n:  27    fib(n):              196418    elapsed seconds: 0.0
n:  28    fib(n):              317811    elapsed seconds: 0.0
n:  29    fib(n):              514229    elapsed seconds: 0.0
n:  30    fib(n):              832040    elapsed seconds: 0.0
n:  31    fib(n):             1346269    elapsed seconds: 0.0
n:  32    fib(n):             2178309    elapsed seconds: 0.0
n:  33    fib(n):             3524578    elapsed seconds: 0.0
n:  34    fib(n):             5702887    elapsed seconds: 0.0
n:  35    fib(n):             9227465    elapsed seconds: 0.0
n:  36    fib(n):            14930352    elapsed seconds: 0.0
n:  37    fib(n):            24157817    elapsed seconds: 0.0
n:  38    fib(n):            39088169    elapsed seconds: 0.0
n:  39    fib(n):            63245986    elapsed seconds: 0.0
n:  40    fib(n):           102334155    elapsed seconds: 0.0
n:  41    fib(n):           165580141    elapsed seconds: 0.0
n:  42    fib(n):           267914296    elapsed seconds: 0.0
n:  43    fib(n):           433494437    elapsed seconds: 0.0
n:  44    fib(n):           701408733    elapsed seconds: 0.0
n:  45    fib(n):          1134903170    elapsed seconds: 0.0
n:  46    fib(n):          1836311903    elapsed seconds: 0.0
n:  47    fib(n):          2971215073    elapsed seconds: 0.0
n:  48    fib(n):          4807526976    elapsed seconds: 0.0
n:  49    fib(n):          7778742049    elapsed seconds: 0.0
n:  50    fib(n):         12586269025    elapsed seconds: 0.0
n:  51    fib(n):         20365011074    elapsed seconds: 0.0
n:  52    fib(n):         32951280099    elapsed seconds: 0.0
n:  53    fib(n):         53316291173    elapsed seconds: 0.0
n:  54    fib(n):         86267571272    elapsed seconds: 0.0
n:  55    fib(n):        139583862445    elapsed seconds: 0.0
n:  56    fib(n):        225851433717    elapsed seconds: 0.0
n:  57    fib(n):        365435296162    elapsed seconds: 0.0
n:  58    fib(n):        591286729879    elapsed seconds: 0.0
n:  59    fib(n):        956722026041    elapsed seconds: 0.0
n:  60    fib(n):       1548008755920    elapsed seconds: 0.0
n:  61    fib(n):       2504730781961    elapsed seconds: 0.0
n:  62    fib(n):       4052739537881    elapsed seconds: 0.0
n:  63    fib(n):       6557470319842    elapsed seconds: 0.0
n:  64    fib(n):      10610209857723    elapsed seconds: 0.0
n:  65    fib(n):      17167680177565    elapsed seconds: 0.0
n:  66    fib(n):      27777890035288    elapsed seconds: 0.0
n:  67    fib(n):      44945570212853    elapsed seconds: 0.0
n:  68    fib(n):      72723460248141    elapsed seconds: 0.0
n:  69    fib(n):     117669030460994    elapsed seconds: 0.0
n:  70    fib(n):     190392490709135    elapsed seconds: 0.0
n:  71    fib(n):     308061521170129    elapsed seconds: 0.0
```

```
n:  72    fib(n):       498454011879264    elapsed seconds: 0.0
n:  73    fib(n):       806515533049393    elapsed seconds: 0.0
n:  74    fib(n):      1304969544928657    elapsed seconds: 0.0
n:  75    fib(n):      2111485077978050    elapsed seconds: 0.0
n:  76    fib(n):      3416454622906707    elapsed seconds: 0.0
n:  77    fib(n):      5527939700884757    elapsed seconds: 0.0
n:  78    fib(n):      8944394323791464    elapsed seconds: 0.0
n:  79    fib(n):     14472334024676221    elapsed seconds: 0.0
n:  80    fib(n):     23416728348467685    elapsed seconds: 0.0
n:  81    fib(n):     37889062373143906    elapsed seconds: 0.0
n:  82    fib(n):     61305790721611591    elapsed seconds: 0.0
n:  83    fib(n):     99194853094755497    elapsed seconds: 0.0
n:  84    fib(n):    160500643816367088    elapsed seconds: 0.0
n:  85    fib(n):    259695496911122585    elapsed seconds: 0.0
n:  86    fib(n):    420196140727489673    elapsed seconds: 0.0
n:  87    fib(n):    679891637638612258    elapsed seconds: 0.0
n:  88    fib(n):   1100087778366101931    elapsed seconds: 0.0
n:  89    fib(n):   1779979416004714189    elapsed seconds: 0.0
n:  90    fib(n):   2880067194370816120    elapsed seconds: 0.0
n:  91    fib(n):   4660046610375530309    elapsed seconds: 0.0
n:  92    fib(n):   7540113804746346429    elapsed seconds: 0.0
n:  93    fib(n):  12200160415121876738    elapsed seconds: 0.0
n:  94    fib(n):  19740274219868223167    elapsed seconds: 0.0
n:  95    fib(n):  31940434634990099905    elapsed seconds: 0.0
n:  96    fib(n):  51680708854858323072    elapsed seconds: 0.0
n:  97    fib(n):  83621143489848422977    elapsed seconds: 0.0
n:  98    fib(n): 135301852344706746049    elapsed seconds: 0.0
n:  99    fib(n): 218922995834555169026    elapsed seconds: 0.0
n: 100    fib(n): 354224848179261915075    elapsed seconds: 0.0
```