Zeitkomplexität

Prof. Dr. Dörte Haftendorn, MuPAD 4, Jan. 07 Update 07
Web: http://haftendorn.uni-lueneburg.de www.mathematik-verstehen.de

++

Diverse Komplextitätsfunktionen **beim Sortieren**:

Stirlingformel

```
lnVon_nMinus1Fakultaet:=n->(n-1/2)*ln(n-1)-n+1+ln(sqrt(2*PI)
```

$$n \to \left(n - \frac{1}{2}\right) \cdot \ln(n-1) - n + 1 + \ln(\sqrt{2 \cdot \pi})$$

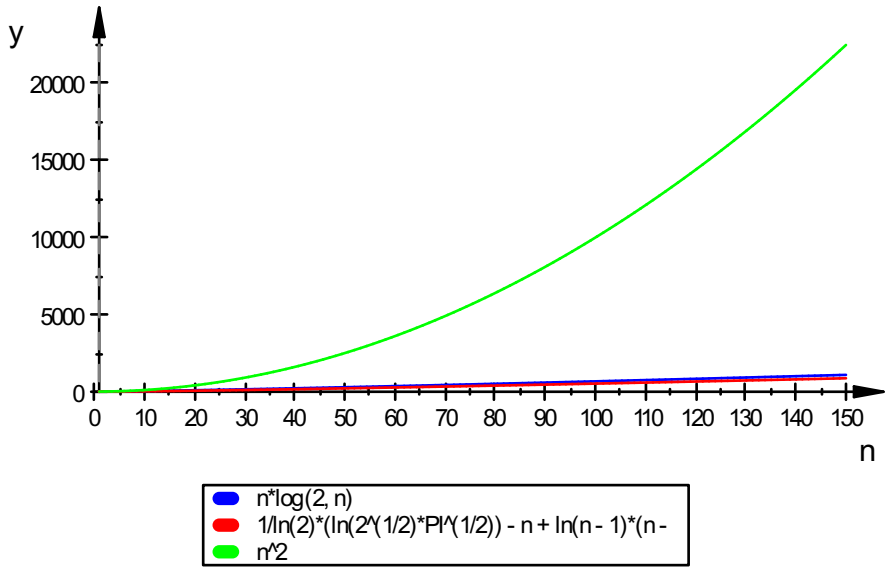
```
[float(lnVon_nMinus1Fakultaet(n)) $ n=10..15]
```

[12.79257202, 15.09608201, 17.49473417, 19.98027166, 22.54575486, 25.1852

```
float(ln((n-1)!)) $n=10..15
```

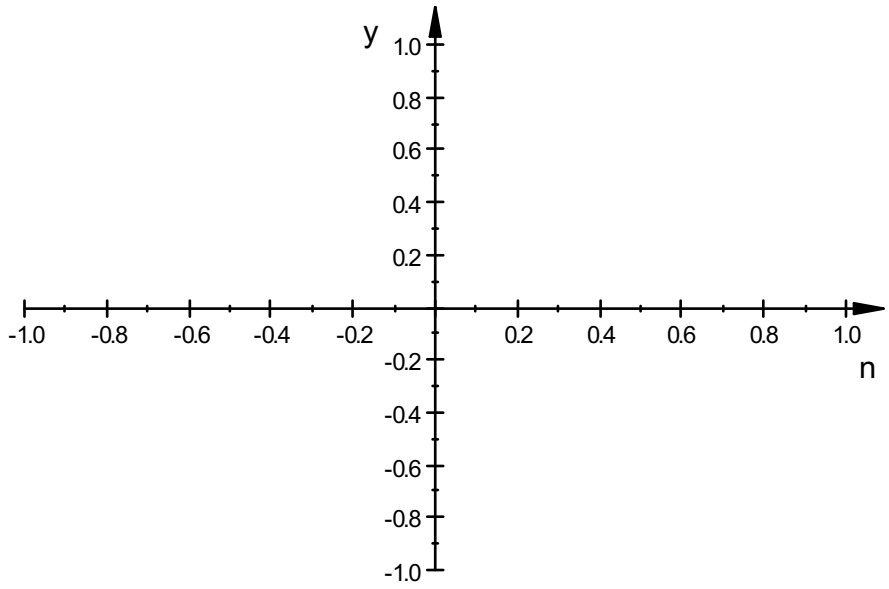
12.80182748, 15.10441257, 17.50230785, 19.9872145, 22.55216385, 25.191221

```
plotfunc2d(n*log(2,n),1/ln(2)*lnVon_nMinus1Fakultaet(n),n^2,
```



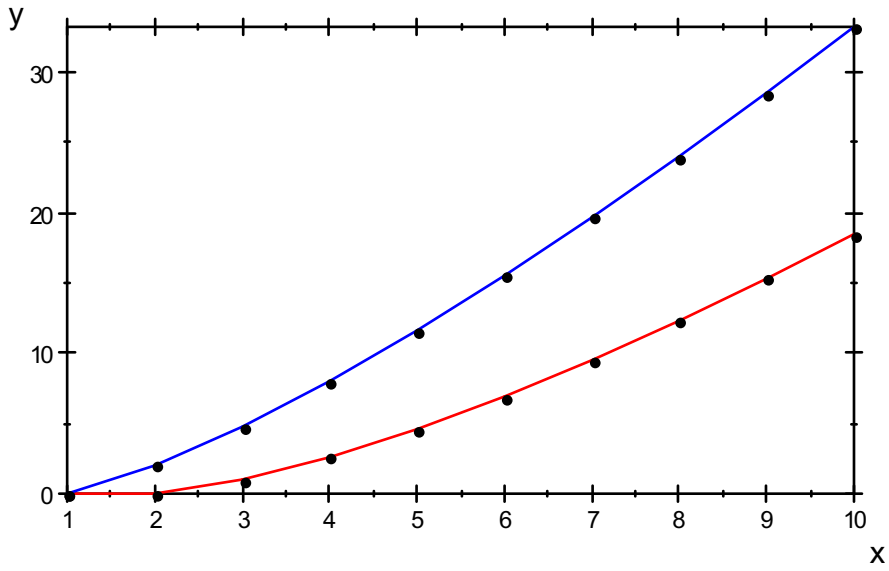
Mit der log(2,(n-1)!) und der nachfolgenden Version klappt es nicht.

```
plotfunc2d(float( eval(1/ln(2)*ln((n-1)!))),n=0..10)
```



nichts zu sehen, daher habe ich oben die Stirlingformel verwendet.
Mit Listen geht es auch mit der Originalformel:

```
liplfak:=plot::Listplot(  [ (log(2,(n-1)!) $ n=1..10) ], Lir
liplnlo:=plot::Listplot( [n*log(2,n) $ n=1..10], LineColor=|
plot(liplfak,liplnlo)
```




```
matrix([float([log(2,(n-1)!),n*log(2,n),n^2]) $ n=1..15])
```

0	0	1.0
0	2.0	4.0
1.0	4.754887502	9.0
2.584962501	8.0	16.0
4.584962501	11.60964047	25.0
6.906890596	15.509775	36.0
9.491853096	19.65148445	49.0
12.29920802	24.0	64.0
15.29920802	28.52932501	81.0
18.46913302	33.21928095	100.0
21.79106111	38.05374781	121.0
25.25049273	43.01955001	144.0
28.83545523	48.10571634	169.0
32.53589495	53.30296891	196.0
36.34324987	58.60335893	225.0

Weitere Komplextiätsfunktionen und Zeiten

```
[1,60,60*60,60*60*24,60*60*24*365,60*60*24*365*3200]
```

[1, 60, 3600, 86400, 31536000, 100915200000]

Anzahl der Sekunden in 1 s,1 min,1 h,1 d, 1a, seit Ramses II

```
sekProJahr:=31.536000*10^6// 31 Millionen
```

31536000.0

```
sekSeit_homo_erectus:=sekProJahr*600000
```

$1.89216 \cdot 10^{13}$

```
10^20/sekProJahr
```

$3.170979198 \cdot 10^{12}$

10^20 s sind 3 Billionen Jahre

```
sekSeitUrknall:=sekProJahr*13*10^9
```

$4.09968 \cdot 10^{17}$

weniger als 10^18 s seit dem Urknall

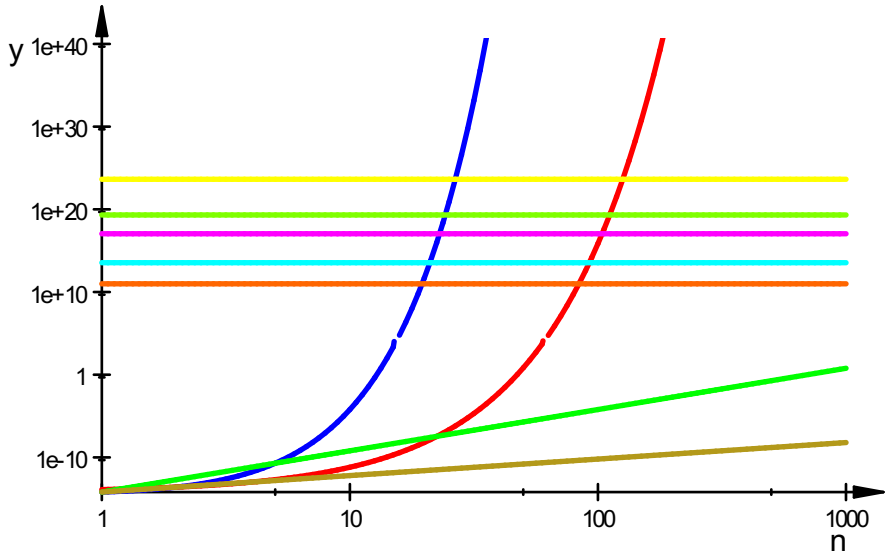
```
minProJahr:=24*365*60
```

525600

Betrachtung der exponentiellen und super-exponentiellen Wachstumsfunktionen gegenüber den polyminialen Wachstumsfunktionen.
Wir betrachten Superrechner (Parallelrechner, Rechnercluster...) mit
200 Teraflops= 200 10^12 Gleitkommaoperationen pro sek.


```
t0:=5*10^(-15):
plotfunc2d(t0*n^n,t0*2^n,t0*n^5,t0*n^2,
```

```
8.6*10^10,31*10^12,10^17,
sekSeit_homo_erectus*10^6,4*10^23,
n=1..1000,ViewingBoxYRange=5*10^-15..10^40 ,
LineWidth=0.8,CoordinateType = LogLog, LegendVisible=F
```



n= Umfang der Eingabe
Ordinate= Zahl der Mikrosekunden Rechenzeit bei 200 10^12 Operationen pro sek, also ein Operation in t0=5 10^-15 s
Waagerechten:Mikrosek pro Tag, Mikrosek pro Jahr, Mikrosek seit Ramses II, Mikrosek seit Homo Erectus, Mikrosek seit Urknall.