**Analyse vom Pascalschen**

**Dreieck**

***Abgabe*** *zum Aufgabenblatt 4*

*vom 18. April 2017*

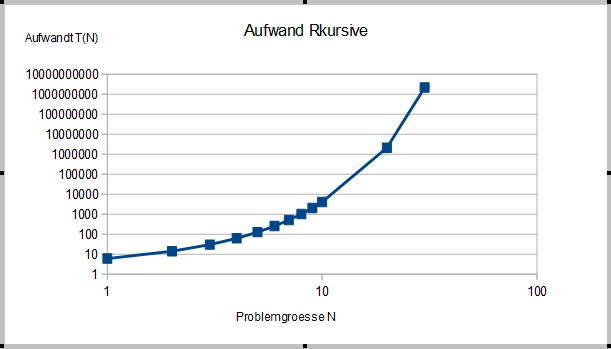
***Karanpal Singh und Johannes Kruber***

# 2 Zusammenhang Problemgröße und Aufwand

|  |  |  |  |
| --- | --- | --- | --- |
| ***Problemgröße N*** | ***Aufwand T(N) For-Schleife*** | ***Aufwand T(N) Rekursive*** | ***Aufwand T(N) Binomial*** |
| 1 | 7 | 6 | 1 |
| 2 | 12 | 14 | 1 |
| 3 | 18 | 30 | 1 |
| 4 | 25 | 62 | 3 |
| 5 | 33 | 126 | 3 |
| 6 | 42 | 254 | 6 |
| 7 | 52 | 510 | 6 |
| 8 | 63 | 1022 | 10 |
| 9 | 75 | 2046 | 10 |
| 10 | 88 | 4096 | 15 |
| 20 | 273 | 2101246 | 55 |
| 30 | 558 | 2149584894 | 120 |
| 40 | 943 |  | 210 |
| 50 | 1428 |  | 325 |
| 60 | 2013 |  | 465 |
| 70 | 2698 |  | 630 |
| 80 | 3483 |  | 820 |
| 90 | 4368 |  | 1035 |
| 100 | 5353 |  | 1275 |

# 3 Grafische Darstellung

## 3.1 Rekursiver Ansatz

*Grafik 1*

Aus der Tabelle kann man erkennen das der Aufwand sich mit jeder Erhöhung der Problemgröße verdoppelt daher wacht der Aufwand Exponentiell wie in Grafik 1 zu erkennen ist.

Aufwand T(N) = O(2^N)

## 3.2For-schleifen Ansatz*Grafik 2*

Bei dem Algorithmus für Die For-Schleifen Lösung handelt es sich einfach um zwei ineinander geschachtelte schleifen, daher ist der Aufwand fast Quadratich. Da die Innere schleife nicht immer Bis zur Problemgröße N läuft ist der auf wand N log N.

Aufwand T(N) = N log N + N = O(N log N)

## 3.3 Binomialer Ansatz*Grafik 3*

Der Algorithmus Verwendet zu Berechnung de n-ten Pascalreihe die Produkt Formel des Binomial Koefizienten: ∏(n-k + i)/i. Das Diagramm zeigt den Aufwand bei zunehmender Pascalreihe.

Aufwand: T(N) = (N/2)2= N2/4=O(N2)