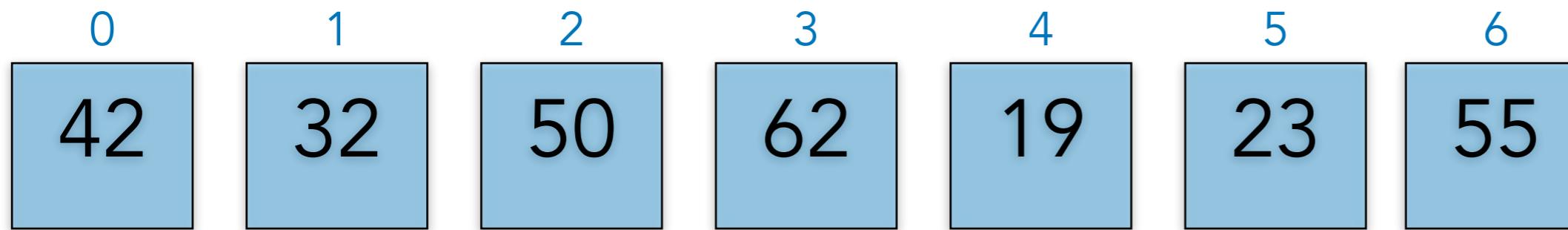


5. Sortieren und Suchen

5.2 Der Bubblesort - Zeitverhalten

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



42 - 32 - 50 - 62 - 19 - 23 - 55 1 V 1 T

32 - 42 - 50 - 62 - 19 - 23 - 55 1 V

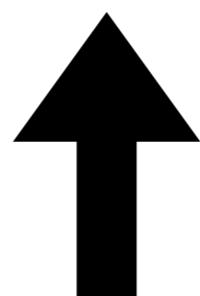
32 - 42 - 50 - 62 - 19 - 23 - 55 1 V

32 - 42 - 50 - 19 - 62 - 23 - 55 1 V 1 T

32 - 42 - 50 - 19 - 23 - 62 - 55 1 V 1 T

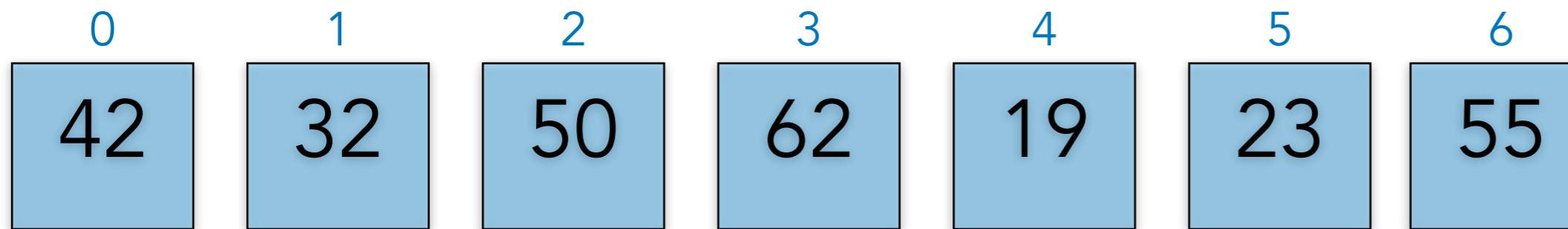
32 - 42 - 50 - 19 - 23 - 55 - 62 1 V 1 T

18 Operationen
im Durchgang 1

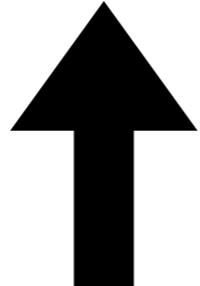


Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort

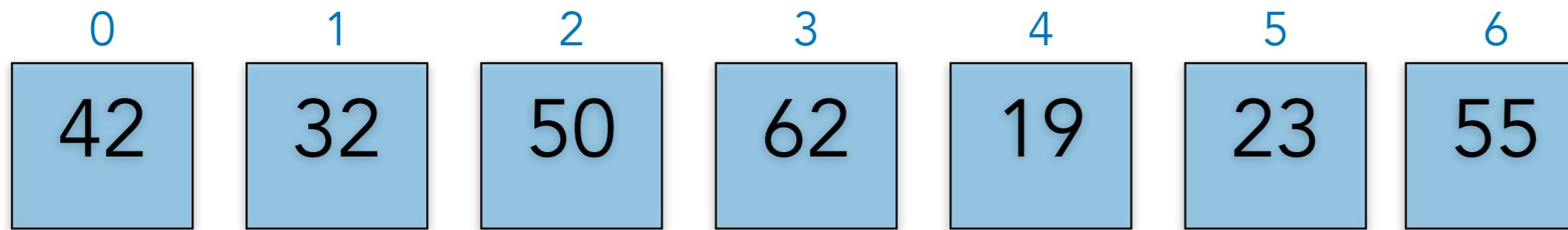


32 - 42 - 50 - 19 - 23 - 55 - 62	1 V		
32 - 42 - 50 - 19 - 23 - 55 - 62	1 V		
32 - 42 - 19 - 50 - 23 - 55 - 62	1 V	1 T	11 Operationen
32 - 42 - 19 - 23 - 50 - 55 - 62	1 V	1 T	im Durchgang 2
32 - 42 - 19 - 23 - 50 - 55 - 62	1 V		

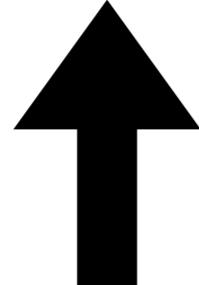


Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort

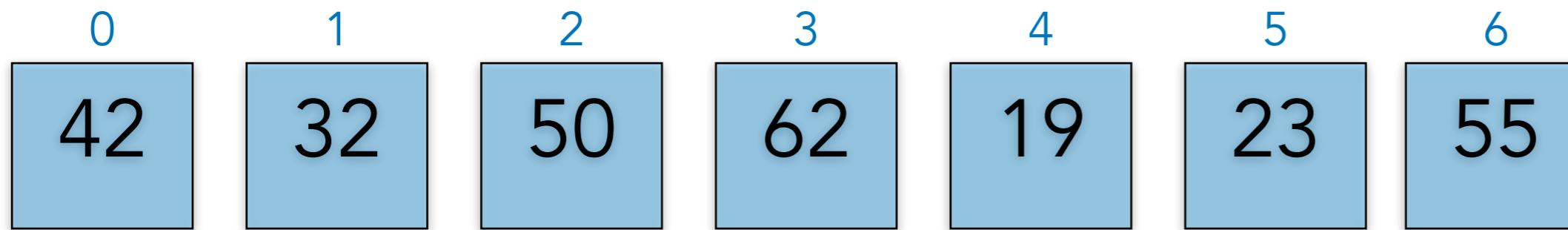


32 - 42 - 19 - 23 - 50 - 55 - 62	1 V		
32 - 19 - 42 - 23 - 50 - 55 - 62	1 V	1 T	
32 - 19 - 23 - 42 - 50 - 55 - 62	1 V	1 T	10 Operationen
32 - 19 - 23 - 42 - 50 - 55 - 62	1 V	im Durchgang 3	

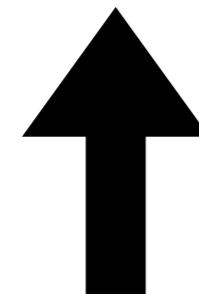


Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



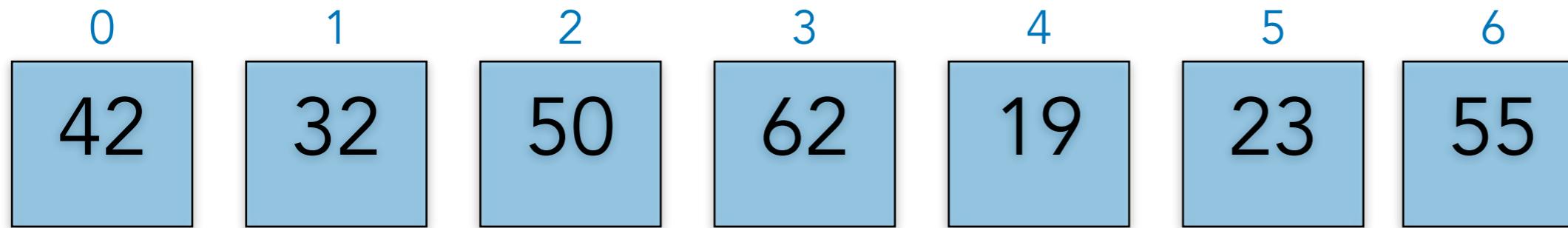
19 - 32 - 23 - 42 - 50 - 55 - 62 1 V 1 T
19 - 23 - 32 - 42 - 50 - 55 - 62 1 V 1 T



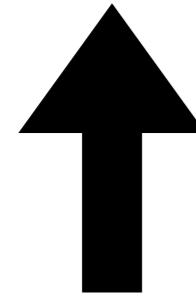
8 Operationen
im Durchgang 4

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



19 - 32 - 23 - 42 - 50 - 55 - 62 1 V



1 Operation
im Durchgang 5

$18 + 11 + 10 + 8 + 1 = 48$ Operationen für $N = 7$ und $N^2 = 49$

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort

```
public void bubblesortMitZaehlung()
{
    int operationen = 0;

    for (int durchgang = 1; durchgang < MAX-1; durchgang++)
    {
        operationen += 3; // init, vergleich, inkrement
        for (int i=0; i<MAX-1; i++)
        {
            operationen += 3; // init, vergleich, inkrement

            if (zahlen[i] > zahlen[i+1])
            {
                tauschen(i , i+1);
                operationen += 4; // tauschen, vergleich
            }
            else
                operationen += 1; // nur vergleichen
        }
    }
}
```

100	Zahlen	46623	Operationen.
200	Zahlen	187872	Operationen.
400	Zahlen	755862	Operationen.
800	Zahlen	3036087	Operationen.
1600	Zahlen	12113724	Operationen.

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort

```
public void bubblesortMitZaehlung()
{
    int operationen = 0;

    for (int durchgang = 1; durchgang < MAX-1; durchgang++)
    {
        operationen += 3; // init, vergleich, inkrement
        for (int i=0; i<MAX-1; i++)
        {
            operationen += 3; // init, vergleich, inkrement

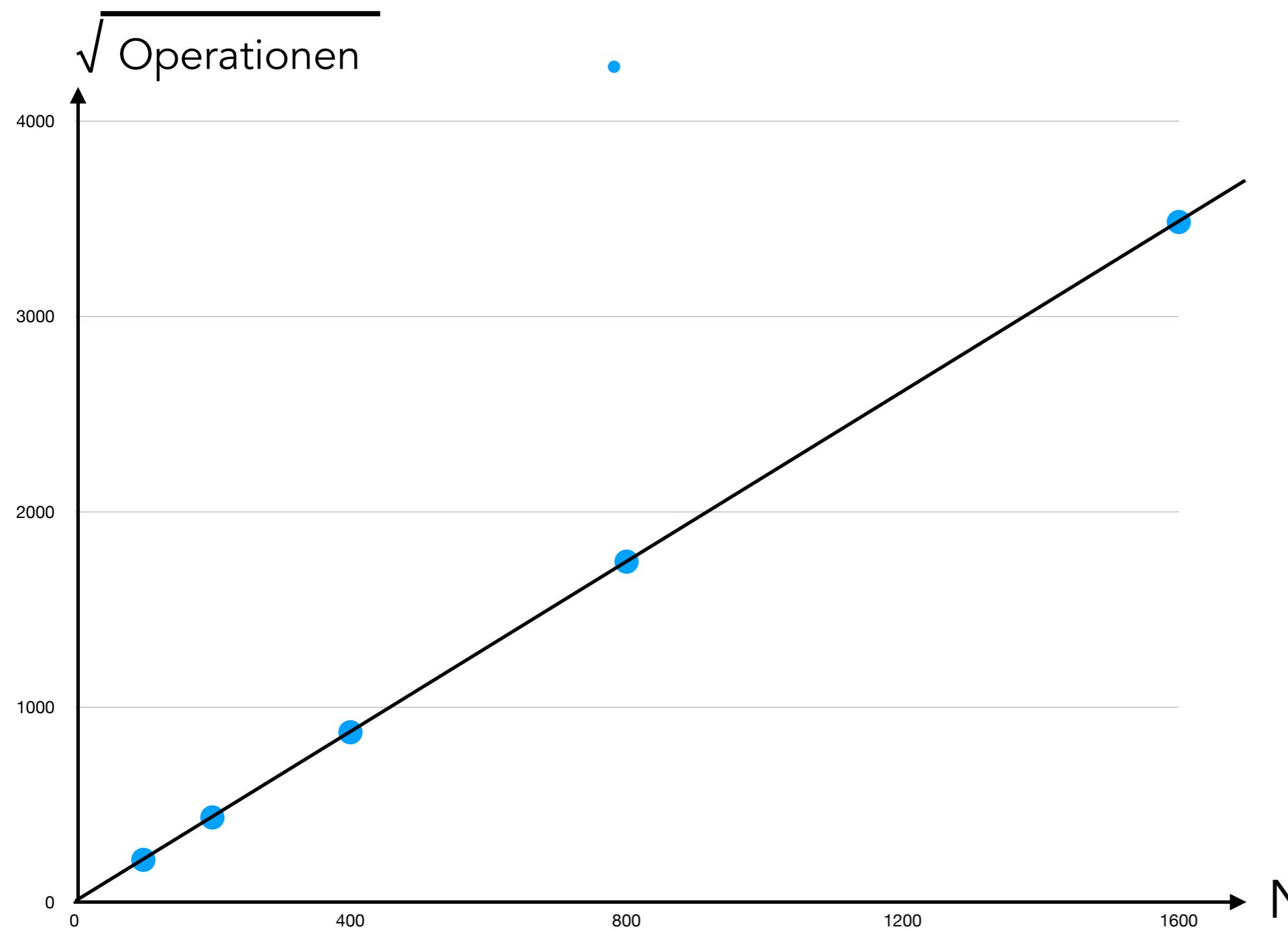
            if (zahlen[i] > zahlen[i+1])
            {
                tauschen(i , i+1);
                operationen += 4; // tauschen, vergleich
            }
            else
                operationen += 1; // nur vergleichen
        }
    }
}
```

100	Zahlen	46623	Operationen.
200	Zahlen	187872	Operationen.
400	Zahlen	755862	Operationen.
800	Zahlen	3036087	Operationen.
1600	Zahlen	12113724	Operationen.

100	Zahlen	216 ²	Operationen.
200	Zahlen	433 ²	Operationen.
400	Zahlen	869 ²	Operationen.
800	Zahlen	1742 ²	Operationen.
1600	Zahlen	3480 ²	Operationen.

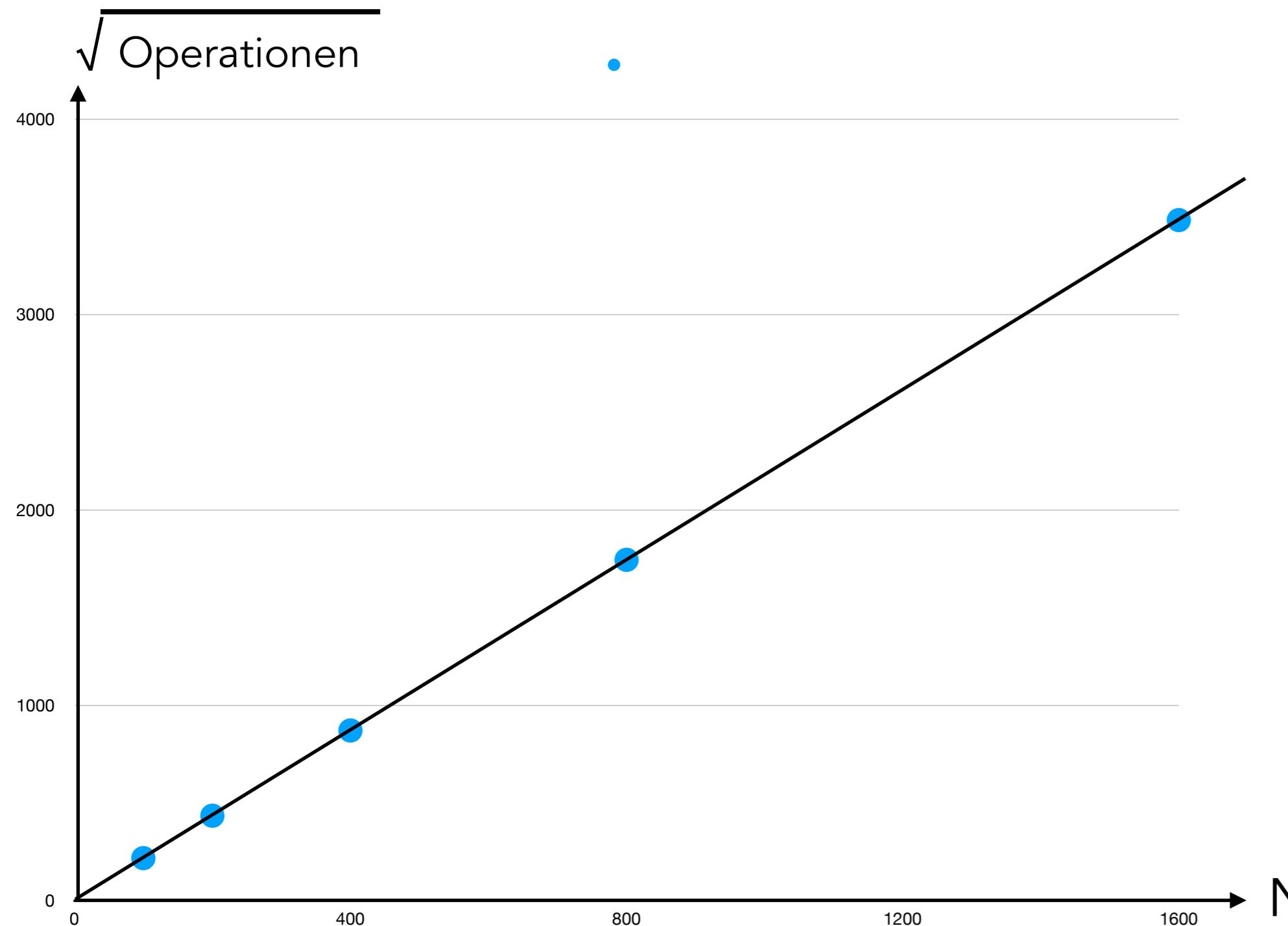
Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



100	Zahlen	216^2 Operationen.
200	Zahlen	433^2 Operationen.
400	Zahlen	869^2 Operationen.
800	Zahlen	1742^2 Operationen.
1600	Zahlen	3480^2 Operationen.

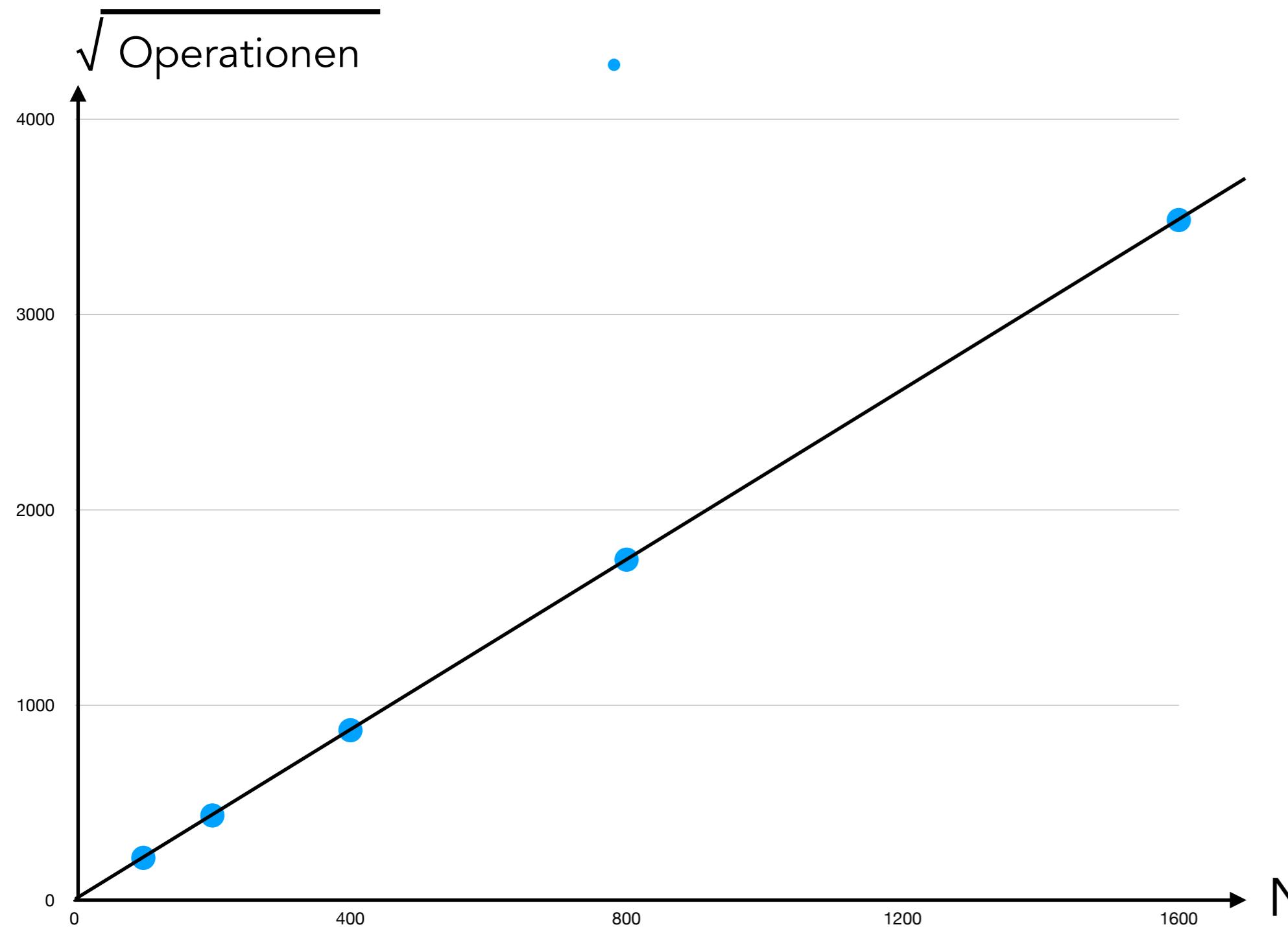
5.2 Bubblesort



Die Quadratwurzel des Rechenaufwands
ist der Zahl der Elemente N
proportional.

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



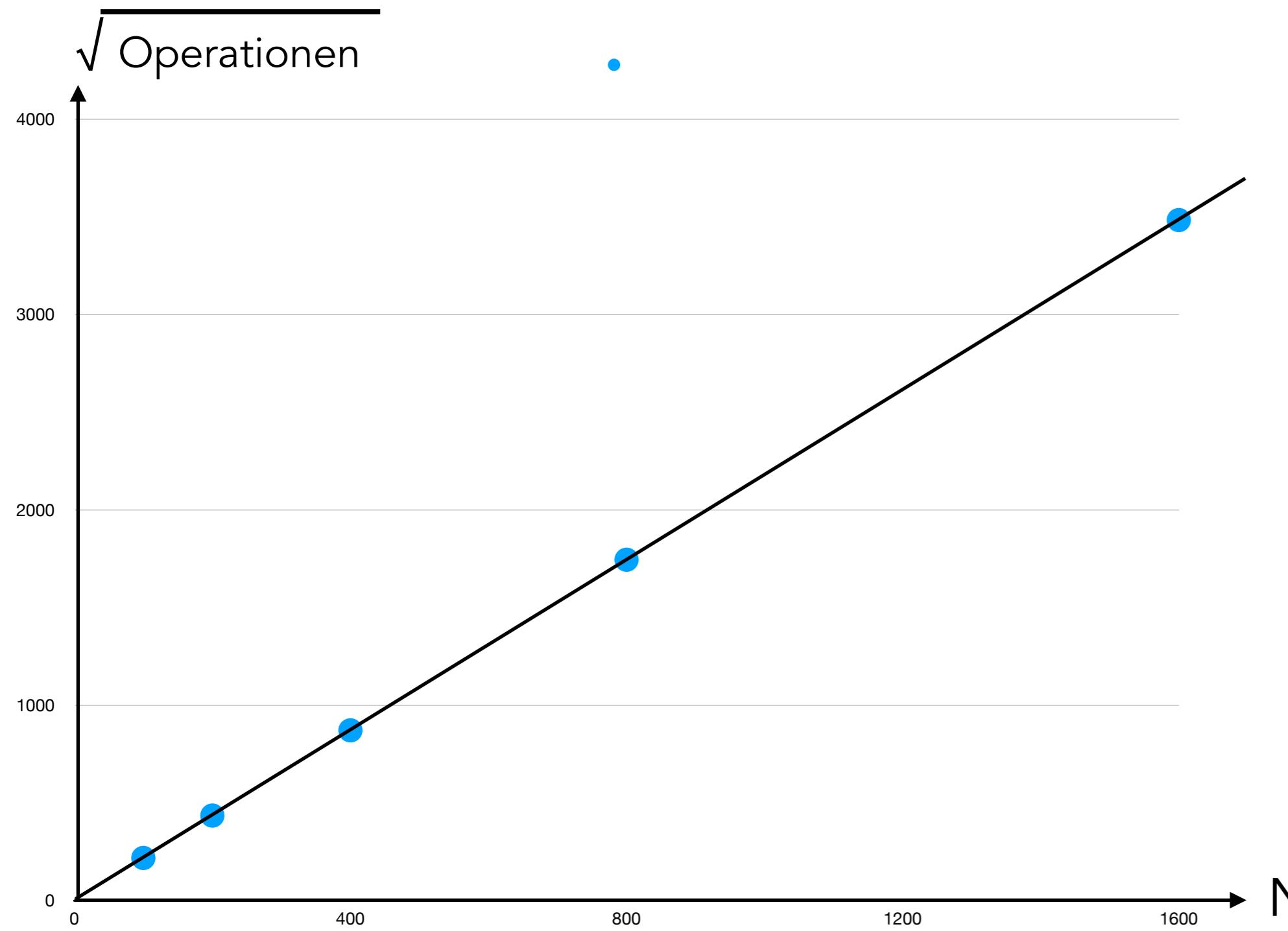
100	Zahlen	216^2 Operationen.
200	Zahlen	433^2 Operationen.
400	Zahlen	869^2 Operationen.
800	Zahlen	1742^2 Operationen.
1600	Zahlen	3480^2 Operationen.

Die Quadratwurzel des Rechenaufwands ist der Zahl der Elemente N proportional. →

Die Zahl der Operationen **O** ist dem Quadrat der Elementzahl **N** proportional.

Einführung in die Objektorientierte Programmierung (OOP)

5.2 Bubblesort



100 Zahlen	216 ² Operationen.
200 Zahlen	433 ² Operationen.
400 Zahlen	869 ² Operationen.
800 Zahlen	1742 ² Operationen.
1600 Zahlen	3480 ² Operationen.

Die Quadratwurzel des Rechenaufwands ist der Zahl der Elemente N proportional. →

Die Zahl der Operationen **O** ist dem Quadrat der Elementzahl **N** proportional. →

$$O = k \cdot N^2 \rightarrow$$

Zeitaufwand = **O(N²)**.