

# Aufgaben: O-Notation

1. Geben Sie die Komplexität folgender Programmabschnitte an.  $n$  ist im Folgenden die Problemgröße.

(a)

```
int a = 5;
int b = 7;
int c = a + b;
int d = n / 2;
```

(b)

```
for( i = 0; i < n; i++)
{
    c += i;
}
```

(c)

```
for( i = 0; i < n; i++)
{
    for( j = 0; j < n; j++)
    {
        c += i * j ;
    }
}
```

(d)

```
int h = 1;
while( h <= n )
{
    c += h;
    h *= 2;
}
```

```
(e)   for( i = 0; i < n; i++)
      {
        for( j = 0; j < i; j++)
          {
            c += i * j ;
          }
      }
```

```
(f)   h = n;
      for( i = 0; i < h; i++)
        {
          for( j = 0; j < i; j++)
            {
              c += i * j ;
            }
          h = h / 2;
        }
```

## 2. Chip-Testing Problem

Schauen Sie sich das Chip-Testing Problem an:

<http://www.cs.mcgill.ca/%7Eecs251/OldCourses/1997/topic3/>

Versuchen Sie es zu verstehen und beantworten Sie die folgenden Fragen:

- Was ist der Best-Case, Worst-Case und der Average-Case.
- Wie sehen die Zeitkomplexitäten für die naive Methode und den Divide and Conquer Algorithmus aus ( O-Notation ).