

4.3.F Datenstrukturen/Arrays – Ergänzungen und Bilder

4.3.F.1 Geschwindigkeit von Algorithmen

Manchmal kann es notwendig sein, einen Programmteil auf Geschwindigkeit zu optimieren. Selbst bei einfachen Abläufen kann es Unterschiede geben. In den folgenden Programmen soll in einem Array nach einem Element mit dem Inhalt 7 gesucht werden:

suchen0.c

```
1 #include <stdio.h>
2 #define ARRSIZE 1000*1000
3 int main(void)
4 {
5     int arr[ARRSIZE]={[200000]=7};
6     int i, fundstelle, gefunden=0;
7     for(i=0; i<ARRSIZE; ++i)
8     {
9         if(arr[i]==7)
10        {
11            gefunden=1;
12            fundstelle=i;
13        }
14    }
15    if(gefunden)
16        printf("Gefunden bei i=%i\n", fundstelle);
17    else
18        printf("Nicht gefunden\n");
19    return 0;
20 }
```

Es ist nicht sinnvoll, weiter nach einem Element zu suchen, wenn man es schon gefunden hat:

suchen1.c

```
1 #include <stdio.h>
2 #define ARRSIZE 1000*1000
3 int main(void)
4 {
5     int arr[ARRSIZE]={[200000]=7};
6     int i, gefunden=0;
7     for(i=0; i<ARRSIZE && gefunden==0; ++i)
8     {
9         if(arr[i]==7)
10        {
11            gefunden=1;
12        }
13    }
14    if(gefunden)
15        printf("Gefunden bei i=%i\n", i-1);
16    else
17        printf("Nicht gefunden\n");
18    return 0;
19 }
```

Hier wird in jedem Schleifendurchlauf dreimal ein Wert abgefragt. Schneller geht es, wenn man nach dem Finden die Laufvariable an das Ende setzt¹. Dann braucht man nur noch zwei Abfragen:

suchen2.c

```
1 #include <stdio.h>
2 #define ARRSIZE 1000*1000
3 int main(void)
4 {
5     int arr[ARRSIZE]={[200000]=7};
6     int i, fundstelle, gefunden=0;
7     for(i=0; i<ARRSIZE; ++i)
8     {
9         if(arr[i]==7)
10        {
11            fundstelle=i;
12            i=ARRSIZE;
13            gefunden=1;
14        }
15    }
16    if(gefunden)
17        printf("Gefunden bei i=%i\n", fundstelle);
18    else
19        printf("Nicht gefunden\n");
20    return 0;
21 }
```

Wenn man nun hinter das Array ein Dummy-Element setzt, das das Suchkriterium erfüllt, kommt man sogar mit einer Abfrage pro Durchlauf aus:

suchen3.c

```
1 #include <stdio.h>
2 #define ARRSIZE 1000*1000
3 int main(void)
4 {
5     int arr[ARRSIZE+1]={[200000]=7, [ARRSIZE]=7};
6     int i, gefunden=0;
7     for(i=0; arr[i]!=7; ++i)
8     {
9     }
10    if(i<ARRSIZE)
11        printf("Gefunden bei i=%i\n", i);
12    else
13        printf("Nicht gefunden\n");
14
15    return 0;
16 }
```

In der Praxis muss man probieren, ob so eine Optimierung hilfreich ist.

¹Alternativ kann man auch auf die Variable `gefunden` verzichten und die Suche im Kopf der Schleife einbauen. Siehe `suche2b.c`