

Grundlegende Suchverfahren in der Informatik

May 25, 2018

1 Sequenzielle Suche

Idee: Durchlaufe die Liste und vergleiche jeden Eintrag mit dem gesuchten Schlüssel x .

Bei einer Liste der Länge n , werden im ungünstigsten Fall n Schlüsselvergleiche nötig und im durchschnittlichen Fall immer noch $n/2$ Schlüsselvergleiche. Der Vorteil der Sequenziellen Suche ist, dass keine Vorsortierung auf der Liste notwendig ist, und die Suche garantiert immer funktioniert.

2 Binäre Suche

In sortierten Listen geht die Suche wesentlich schneller als sequenziell.

Die Idee hinter der Binären Suche ist die rekursive binäre Zerlegung der Liste.

Algorithmus:

```
Suche  $x$  in Liste  $L$ :  
    if ( $L$  empty)  
        report "nicht gefunden";  
    else  
         $mid = \text{floor}(L.length / 2)$ ;  
        if ( $x == L[mid]$ )  
            report "gefunden"  
        if ( $x < L[mid]$ ) Suche  $x$  in Subliste  $L[0..mid-1]$ ;  
        if ( $x > L[mid]$ ) Suche  $x$  in Subliste  $L[mid+1..n-1]$ 
```

Durch rekursive Zerlegung der Liste in zwei Teile sind bei einer Liste der Länge n durchschnittlich nur noch $\log(n)$ Schlüsselvergleiche notwendig, auch wenn der gesuchte Schlüssel x nicht in der Liste ist.

3 Sprungsuche

¹ Bedingung ist immer noch, dass die Liste sortiert ist.

Zerteile die Liste in gleichgroße Stücke der Länge n . Überprüfe in welchem Abschnitt der Suchschlüssel x liegen muss und durchsuche diesen Abschnitt anschließend sequenziell oder binär.

¹https://en.wikipedia.org/wiki/Jump_search

4 Weitere Verfahren

Interpolationssuche²

Exponentielle Suche³

²https://en.wikipedia.org/wiki/Interpolation_search

³https://en.wikipedia.org/wiki/Exponential_search