

Einführung in die Informatik I

Kapitel II.2: Spezielle Suchalgorithmen

Prof. Dr. Marcin Grzegorzek¹

Research Group for Pattern Recognition
www.pr.informatik.uni-siegen.de

Institute for Vision and Graphics
University of Siegen, Germany



¹Die im Rahmen dieser Lehrveranstaltung verwendeten Lernmaterialien wurden uns zum Großteil von Herrn Prof. Dr. Wolfgang Wiechert und Herrn Prof. Dr. Roland Reichardt zur Verfügung gestellt.

Inhaltsverzeichnis

I. MATLAB-Einführung

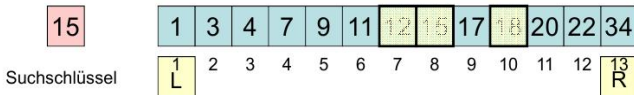
1. Voraussetzungen und Konventionen
2. Variablen und arithmetische Ausdrücke
3. Automatisierungen von Berechnungen
4. Logische Ausdrücke
5. Verzweigungen
6. Schleifen
7. Fehlersuche in Programmen
8. Funktionen
9. Arbeitsweise von Funktionen
10. Vektoren
11. Matrizen

II. Algorithmen

1. Suchen
- ▶ 2. Spezielle Suchalgorithmen
3. Sortieren
4. Rekursion und Quicksort

Binäre Suche (Bi-Sektion)

- Eine effiziente Suchmethode bei **sortierten** Daten ist das Bi-Sektionsverfahren (Divide-and-Conquer).
- Vorgehensweise:
 - Erstelle zwei Grenzen (Links=1, Rechts=n)
 - Prüfe das Element in der Mitte des Feldes, ob es größer oder kleiner als das gesuchte Element ist
 - Verschiebe die linke bzw. rechte Grenze um eins über die Mitte
 - Abbruch: Rechts < Links oder Mitte == Suchschlüssel



Binäre Suche: Details

15

- $Mitte = \text{floor}((\text{Links} + \text{Rechts}) / 2)$; Suchschlüssel

$$M = (13 + 1) / 2 = 7$$

$$15 < 12 \rightarrow L = M + 1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1 L	2	3	4	5	6	7	8	9	10	11	12	13 R

$$M = (13 + 8) / 2 = 10.5$$

$$15 < 18 \rightarrow R = M - 1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8 L	9	10	11	12	13 R

$$M = (9 + 8) / 2 = 8.5$$

$$15 == 15 \rightarrow \text{Gefunden!}$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8 L	9 R	10	11	12	13

Algorithmus: Binäre Suche

- Die Mitte zwischen der linken und rechten Grenze wird durch folgende Formel berechnet (Mittelwert):
 - $\text{Mitte} = \text{floor}((\text{Rechts} + \text{Links}) / 2);$

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8	9	10	11	12	14
							L		R			

- Problematisch bei diesem Algorithmus sind folgende Punkte:
 - Was passiert an der linken Grenze?
 - Was passiert an der rechten Grenze?
 - Was passiert, wenn der Suchschlüssel gar nicht in dem Feld vorhanden ist?

Binäre Suche: Linke Grenze

1

Suchschlüssel

$$M=(13+1)/2=7$$

$$1 < 12 \rightarrow R=M-1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1 L	2	3	4	5	6	7	8	9	10	11	12	13 R

$$M=(6+1)/2=3.5$$

$$1 < 4 \rightarrow R=M-1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1 L	2	3	4	5	6 R	7	8	9	10	11	12	13

$$M=(1+2)/2=1.5$$

$$1 == 1 \rightarrow \text{Gefunden!}$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1 L	2 R	3	4	5	6	7	8	9	10	11	12	13

Binäre Suche: Rechte Grenze

34

Suchschlüssel

$$M=(1+13)/2=7$$

$$34 < 12 \rightarrow L=M+1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
---	---	---	---	---	----	----	----	----	----	----	----	----

1 L	2	3	4	5	6	7	8	9	10	11	12	13 R
--------	---	---	---	---	---	---	---	---	----	----	----	---------

$$M=(8+13)/2=10.5$$

$$34 < 18 \rightarrow L=M+1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
---	---	---	---	---	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8 L	9	10	11	12	13 R
---	---	---	---	---	---	---	--------	---	----	----	----	---------

$$M=(11+13)/2=12$$

$$34 < 22 \rightarrow L=M+1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
---	---	---	---	---	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8	9	10	L	12	13 R
---	---	---	---	---	---	---	---	---	----	---	----	---------

$$M=(13+13)/2=13$$

$$34 == 34 \rightarrow \text{Gefunden!}$$

1	3	4	7	9	11	12	15	17	18	20	22	34
---	---	---	---	---	----	----	----	----	----	----	----	----

1	2	3	4	5	6	7	8	9	10	11	12	13 LR
---	---	---	---	---	---	---	---	---	----	----	----	----------

Binäre Suche: Erfolglos

13

Suchschlüssel

$$M=(13+1)/2=7$$

$$13 < 12 \rightarrow L=M+1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8	9	10	11	12	13
L												R

$$M=(13+8)/2=10.5$$

$$13 < 18 \rightarrow R=M-1$$

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8	9	10	11	12	13
							L					R

$$M=(9+8)/2=8.5$$


$$13 < 15 \rightarrow R=M-1$$

$L \leq R \rightarrow$ **Nicht gefunden!**

1	3	4	7	9	11	12	15	17	18	20	22	34
1	2	3	4	5	6	7	8	9	10	11	12	13
							L	R				

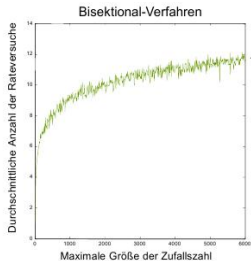
Binäre Suche: Algorithmus

```
function Position=Suche_BiSec(x,Key)
Links    = 1;                % Erste Position
Rechts   = length(x);       % Letzte Position
Position = 0;                % Default
while Links<=Rechts
    Mitte = floor((Links+Rechts)/2);
    if     x(Mitte)==Key      % gefunden..
        Position = Mitte;
        break;
    elseif x(Mitte)< Key     % nicht gefunden...
        Links    = Mitte+1;
    else
        Rechts   = Mitte-1;
    end;
end;
```

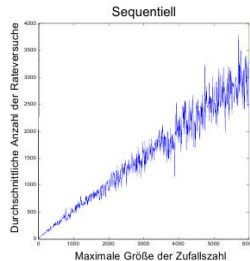


Analyse des Laufzeitverhaltens

- Je 30 mal eine Zufallszahl im Intervall 1:n zu finden.
- Die Mittelwert der 30 Versuche wird gebildet



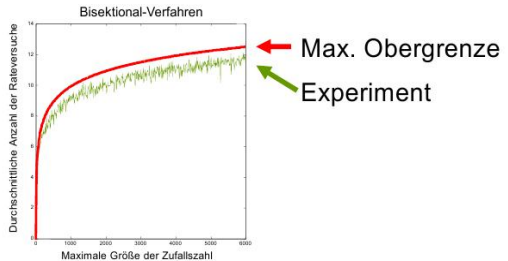
← 12



← 3000

Eigenschaften: Binäre Suche

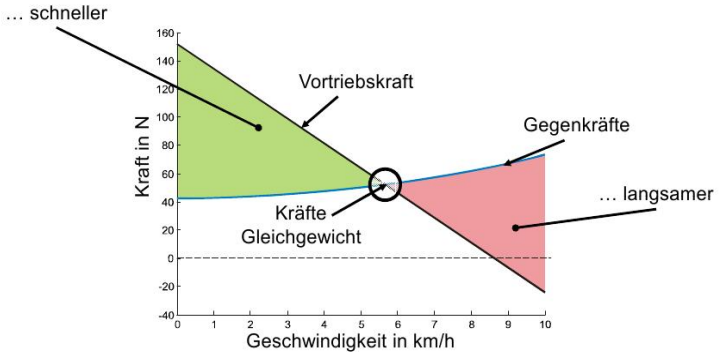
- Die binäre Suche erfordert niemals mehr als $\log_2(n+1)$ Vergleiche
- Die Eigenschaft folgt unmittelbar aus der Tatsache, dass das Suchintervall **wenigstens halbiert** wird.



Zusammenfassung: Suchen

Zugriff	Zufällig	Sequentiell	Bi-Sektion
Findet Schlüssel, wenn in der Liste	Ja	Ja	Ja
Abbruch, wenn nicht in der Liste	Nein	Ja	Ja
Voraussetzung: sortiert	Nein	Nein	Ja
Durchschnittliche Zahl der Versuche	n	$n/2$	$\log_2(n)$

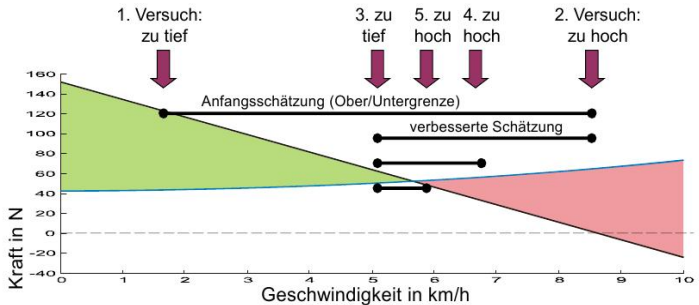
Fahrradbeispiel: Diagramm der Kräfte



- Am Kreuzungspunkt sind die Kräfte gleich groß:

$$F_{Vortrieb} = F_{Roll} + F_{Steigung} + F_{Luft}$$

Einschachteln des Gleichgewichts



- Das Kräftegleichgewicht findet man mit Hilfe einer Variante der binären Suche.
- Allerdings kann man das Ergebnis nur bis auf eine gewisse Genauigkeit ϵ bestimmen.

MATLAB-Implementierung

```
% Eingaben
i =input ('Übersetzung :');
S =input ('Steigung :');
LG =input ('Linke Grenze:');
RG =input ('Rechte Grenze:');
eps=input ('Genauigkeit :');

% Anfangsberechnungen
F_LG=F_Ges (LG,S,i);
F_RG=F_Ges (RG,S,i);

% Anfangsberechnungen
F_LG=F_Ges (LG,Steig,i);
F_RG=F_Ges (RG,Steig,i);

% Fehlerabfrage
if F_LG<0 | F_RG>0
    error('Falsches Anfangsintervall');
else
    while RG-LG>epsilon
        % Mittelwert
        v_M=(RG+LG)/2 ;
        F_M=F_Ges (v_M,Steig,i);

        % Neue Intervallgrenzen
        if F_M>0
            LG =v_M; F_LG=F_M;
        else
            RG =v_M; F_RG=F_M;
        end
    end

    % Ergebnis
    disp (,'Ergebnis zwischen'); LG
    disp ('und'); RG
end
```