

## Info2 Uebung vom 16.4.02

Gruppe F, Thomas Briner, <http://n.ethz.ch/student/brinerth/info2>

15. April 2002

### 1 Laufzeitanalyse: Teleskopieren und Induktionsbeweis

Bsp: Algorithmus zum Finden des Maximums in ungeordnetem Array

Idee: Divide and conquer

- Array immer halbieren
- in jeder Hälfte Maximum bestimmen
- beim Zusammenfügen grösseres der beiden Maxima als neues Maximum speichern

Programmcode

```
MODULE MaxFindenRek;  
  IMPORT Out, RN:=RandomNumbers;  
  CONST LEN=32;  
  
  PROCEDURE vgl(  

```

```
    END vgl;  
  
  PROCEDURE init(VAR array:ARRAY OF REAL);  
    VAR i:INTEGER;  
    BEGIN  
      FOR i:=0 TO LN-1 DO  
        array[i]:=RN.Uniform();  
      END;  
    END init;  
  
  PROCEDURE start*();  
    VAR array:ARRAY LEN OF REAL; res:REAL; nOfVgl:LONGINT;  
    BEGIN  
      init(array);  
      nOfVgl:=0;  
      res:=vgl(array,0,LEN-1,nOfVgl);  
      Out.String("Anzahl Vergleiche: ");  
      Out.Int(nOfVgl,10); Out.Ln;  
      Out.String("Das Maximum ist: ");  
      Out.Real(res,20); Out.Ln;  
    END start;  
  
END MaxFindenRek.  
  
MaxFindenRek.start~  
System.Free MaxFindenRek~
```

Rekursive Gleichung

Wieviele Vergleiche sind das für ein Array mit 4048 Einträgen?  
In welcher Ordnung liegt das?  
→ geschlossene Form wird benötigt

Schritt 1: Teleskopieren  $\approx$  "systematisches Raten"

Schritt 2: Mit vollständiger Induktion die Vermutung beweisen