Sortieren durch Einfügen rekursiv

Sortieren durch Einfügen
schrittweise entwickelt

Sortieren durch Einfügen rekursiv

Voranalyse:

- Die Aufgabe lässt sich in zwei Teilfunktionen aufteilen
  - sortiere eine Zahl in eine sortierte Liste ein
  - bearbeite (damit) die ganze unsortierte Liste
- Funktionsköpfe festlegen, Rümpfe vorbereiten
- Beispiellisten und Tests definieren

Sortieren durch Einfügen rekursiv

```
## Vorgabe der zu sortierenden Liste
zahlen = [30,10,50,5,200,20,120,10,40, 1]

## Modularisierung und Funktionsköpfe
def sortiere_eine_ein( zahl, liste ):
    return 'undefiniert'

def sortiere( liste, sortierte ):
    return 'undefiniert'
### Tests:
print( sortiere_eine_ein(30, [10,40]) )
print( sortiere(zahlen, []) )
```

Diese Liste muss sortiert sein

Sortieren durch Einfügen rekursiv

Schritt 1:

- Elementarfälle behandeln
  - Die vorsortierte Liste (in *sortiere_eine_ein*), in die das eine Element eingeordnet werden soll, kann leer sein.
  - Die Liste der unsortierten Elemente (in *sortiere*) kann leer sein

Sortieren durch Einfügen rekursiv

```
def sortiere_eine_ein( zahl, liste ):
    ## Elementarfall behandeln (hier nicht trivial)
    if len(liste)==0:
        return [zahl]
    return 'undefiniert'

def sortiere( liste, sortierte ):
    ## Elementarfall behandeln
    if len(liste)==0:
        return sortierte
    return 'undefiniert'
```

Sortieren durch Einfügen rekursiv

Schritt 2:

- Ein weiterer Elementarfall
  - Das neue Element ist an der richtigen Stelle.

Ein Test dazu:

sortiere_eine_ein(5, [10,40])

Sortieren durch Einfügen rekursiv

```
def sortiere_eine_ein( zahl, liste ):
    if len(liste)==0:
        return [zahl]
    ## zweiten Elementarfall behandeln!
    if zahl<liste[0]:
        return [zahl]+liste
    return 'undefiniert'

def sortiere( liste, sortierte ):
    if len(liste)==0:
        return sortierte
    return 'undefiniert'
```

Sortieren durch Einfügen rekursiv

Schritt 3:

- Normalfall in sortiere_eine_ein
  - In der Restliste weiter die richtig Position suchen
  - In der Rückgabe die Liste wieder aufbauen

Sortieren durch Einfügen rekursiv

```
def sortiere_eine_ein( zahl, liste ):
    if len(liste)==0:
        return [zahl]
    if zahl<liste[0]:
        return [zahl]+liste
    ## "Normalfall" behandeln
    return liste[:1] + sortiere_eine_ein( zahl, liste[1:] )

def sortiere( liste, sortierte ):
    if len(liste)==0:
        return sortierte
    return 'undefiniert'
```

Sortieren durch Einfügen rekursiv

Schritt 4:

- Normalfall in sortiere
  - Die Restliste weiter bearbeiten mit der bearbeiteten Liste sortierte, in die das aktuell erste Element der unsortierten Liste durch sortiere_eine_ein eingefügt wird.

Sortieren durch Einfügen rekursiv

```
def sortiere_eine_ein( zahl, liste ):
    if len(liste)==0:
        return [zahl]
    if zahl<liste[0]:
        return [zahl]+liste
    return liste[:1] + sortiere_eine_ein( zahl, liste[1:] )

def sortiere( liste, sortierte ):
    if len(liste)==0:
        return sortierte
    ## "Normalfall" behandeln
    return sortiere( liste[1:],
                     sortiere_eine_ein(liste[0], sortierte) )
```