

Info2 Uebung vom 2.4.02

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

21. April 2002

1 Nachbesprechung Uebung 1

zu 1.1:

Die meisten haben sich für diese Lösung entschieden:

$$\Theta(g(n)) := \{f : N \rightarrow R^+ : \exists c > 0, n_0 \in N : \forall n > n_0 : f(n) = c * g(n)\}$$

Bsp:

$f(n) = 3n^2$, $g(n) = n^2 + 5n$. Hier gilt, dass $f(n) \in O(g(n))$ und $g(n) \in O(f(n))$.
Finde ein c , so dass die Gleichung erfüllt ist!

zu 1.2

Rezept für Vorgehen:

1. c wählen
2. mögliche n aus der Gleichung bestimmen
3. n_0 wählen, indem der positive Wert aufgerundet wird.

Dieses Vorgehen ist grundsätzlich richtig, aber was geht hier schief?

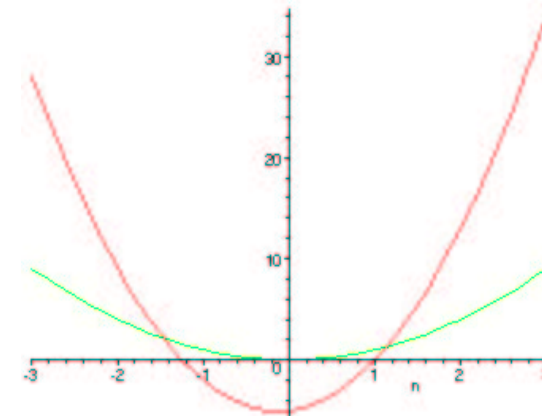
$$4n^2 + n - 5 \leq c * n^2$$

$c:=1$ wählen

$$3n^2 + n - 5 \leq 0$$

$$n_1, n_2 = \frac{-1 \pm \sqrt{1 + 4 * 3 * 5}}{6} = \frac{-1 \pm \sqrt{61}}{6} \cong 1.135, -1.468$$

$c=1, n_0 = 2$



Mit diesem Vorgehen werden allfällige Schnittpunkte der Funktionen ermittelt. Ob das Resultat sinnvoll ist, muss separat überlegt werden, d.h. welche Kurve oberhalb von welcher ist.
 $\Rightarrow n^2$ kann für grosse n nicht stärker steigen als $4n^2$

$$\lim_{n \rightarrow \infty} \frac{4n^2 + n - 5}{n^2} = \lim_{n \rightarrow \infty} \left(4n + 1 - \frac{5}{n}\right) \Rightarrow \infty$$

zu 1.4b)

Je nach Zusammenhang/Vorlesung muss die Basis des Logarithmus anders verstanden werden. In der Informatik ist meist der Logarithmus zur Basis 2 gemeint

zu 1.6

$$\text{competitive ratio} = \frac{\text{Algorithmus}_{\text{eigener}}}{\text{Algorithmus}_{\text{idealer}}}$$

2 Client/Servermodule

Ein Clientmodul soll Funktionalitäten, die von einem Servermodul zur Verfügung gestellt werden, benutzen können, ohne die Implementation zu kennen. Es muss einzig die Schnittstelle (Application Interface API) bekannt sein, um genau zu wissen, welche Typen, Variablen und Prozeduren für das Clientmodul sichtbar sind. \Rightarrow Information Hiding, Kapselung

Ziel: Ein Modul, das eine bestimmte Reihe hochzählen kann (zB. 3, 6, 9, 12, ...)

Annahme: Gegeben ein Modul, das folgende Funktionalität anbietet (in Oberon mit Browser.ShowDef anzeigen lassen)

```
DEFINITION ReihenIterator;

    TYPE
        Reihe = POINTER TO ReihenDesc;

    PROCEDURE currentItem (reihe: Reihe): INTEGER;
    PROCEDURE init (start: INTEGER): Reihe;
    PROCEDURE isDone (reihe: Reihe): BOOLEAN;
    PROCEDURE next (reihe: Reihe);

END ReihenIterator.
```

Mit Hilfe dieses Servermoduls wird nun das Aufzählen einer Reihe so durchgeführt:

```
MODULE ReihenClient;

    IMPORT It:=ReihenIterator, Out, In;

    PROCEDURE do*;
        VAR start: INTEGER; reihe: It.Reihe;
        BEGIN
            In.Open;
            In.Int(start);
            IF (In.Done) THEN
                reihe:= It.init(start);
                WHILE (~It.isDone(reihe)) DO
                    Out.Int(It.currentItem(reihe), 5); Out.Ln;
                    It.next(reihe);
                END;
                Out.String("Ende der Reihe"); Out.Ln;
            ELSE
                Out.String("INTEGER als Eingabe");
            END;
        END do;

END ReihenClient.

ReihenClient.do 5~
System.Free ReihenClient~
System.Free ReihenIterator~
```

Implementierung des Servermoduls:

```
MODULE ReihenIterator;
```

```
  CONST LEN=32;
```

3 Uebungsblatt 3

Die Uebung 3.2 ist das Erstellen des Servermoduls. In 3.3 wird dann ein Clientmodul zu diesem StackServer geschrieben, der die angebotene Funktionalität benutzt.

Definition der Schnittstelle: