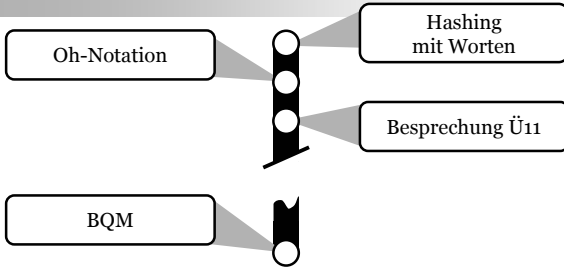


- Letzte Gelegenheit für das Testat

Was wir heute tun




---

---

---

---

---

---

---

---

Hashing mit Worten



```

Jeder Buchstabe und Schriftzahl
ist eine Zahl zwischen 0 und 255
zahl := ORD( buchstabe );
Durch den ARRAY OF CHAR düsen: letztes Zeichen: 0x
Datei öffnen;
erstes zeichen lesen;
total := 0;
WHILE zeichen # „0x“ DO
  REPEAT
    wert := ORD(zeichen)
    total := total + wert; Lies nächstes Zeichen
  UNTIL zeichen = „#“;
  Lies nächstes Zeichen;
END;
RETURN hash( total );
    
```

---

---

---

---

---

---

---

---

Oh-Notation



die Oh-Notation wird verwendet, um asymptotische Laufzeiten von Algorithmen anzugeben

für eine Funktion  $g$ :  $f(n) = O(g(n))$  gilt

$$O(g(n)) \quad \{ f: \quad +: c > 0, n_0 \quad : n \geq n_0: f(n) \leq cg(n) \}.$$

Beispiel  $f(n) = 2n^2 + 3n + 2$

Vermutung  $f$  liegt in  $O(n^2)$

Finde  $c$  und  $n_0$ , so dass  $n \geq n_0: f(n) \leq cn^2$ :

Rate  $c = 3$  und berechne, für welche  $n, f(n) \leq 3n^2$  gilt

$$2n^2 + 3n + 2 \leq 3n^2 \quad n^2 - 3n - 2 \geq 0 \quad n \geq 1.5 + \sqrt{17}/2$$

Also zum Beispiel  $n_0 = 4$

---

---

---

---

---

---

---

---

## Besprechung Übungsserie 11 – Deklarationen



```
TYPE Command = POINTER TO RECORD
  next: Command; END;
TYPE Go = POINTER TO RECORD(Command);
  dist: REAL; END;
TYPE Line = POINTER TO RECORD(Command);
  dist: REAL; END;
TYPE Turn = POINTER TO RECORD(Command);
  angle: REAL; END;
TYPE Color = POINTER TO RECORD(Command);
  color: INTEGER; END;
TYPE Call = POINTER TO RECORD(Command);
  callee: ARRAY MAXNAMELENGTH OF CHAR;
  scale: REAL; END;
TYPE Program = POINTER TO RECORD;
  next: Program;
  first, last: Command;
  name: ARRAY MAXNAMELENGTH OF CHAR; END;

VAR firstprog, lastprog: Program;
```

---

---

---

---

---

---

---

---

## Besprechung Übungsserie 11 – Prozeduren



```
PROCEDURE CreateProgram(): Program;
  (* erstellt neues leeres Programm *)

PROCEDURE AddCommandToProg (p: Program; c: Command);
  (* haengt ein Befehl in ein Programm *)

PROCEDURE ReadCom(VAR s: Texts.Scanner): Command;
  (* parst naechstes Token und erstellt Command *)

PROCEDURE ReadDecl(VAR s: Texts.Scanner): Program;
  (* parst Code und erstellt neues Programm *)

PROCEDURE ReadGraphic(VAR s: Texts.Scanner);
  (* parst Code nach Deklarationen *)

PROCEDURE Execute*(VAR decl: ARRAY OF CHAR;
  scale: REAL);
  (* sucht in geladenen Programmen nach decl und
  fuehrt das Programm mit Skalierung scale aus *)
```

---

---

---

---

---

---

---

---

## Besprechung Übungsserie 11 – Teil 3



```
PROCEDURE ReadCommand(VAR s: Texts.Scanner): Command
VAR olds: Texts.Scanner; c: Command; go: Go;
turn: Turn; line: Line; color: Color; call: Call;
BEGIN
  olds := s;
  Texts.Scan(s);
  IF s.eot THEN
    c := NIL;
  ELSIF s.s = "go" THEN
    NEW(go); Texts.Scan(s); go^.dist := GetReal(s);
    c := go;
  ...
  ELSE
    s := olds; c := NIL;
  END;
  IF c # NIL THEN c^.next := NIL; END;
  RETURN c;
END ReadCommand;
```

---

---

---

---

---

---

---

---