

Musterlösung der 4. Übung

1. Der Algorithmus ist statisch finit (endliche Anzahl von Instruktionen), dynamisch finit (er benötigt eine begrenzte Menge von Variablen, Speicher und sonstigen Ressourcen), und er terminiert (stets nach 100 Schleifendurchläufen).

2. 2 Fälle, die beide zur Aufgabenstellung passen:

Beispiel 1:

```
i := 100;
WHILE TRUE DO
  i := i - 10;
```

Beispiel 2:

```
i := 100;
WHILE i >= 0 DO
  i := i - 10;
```

Beispiel 1: Statisch finit, dynamisch finit, terminiert nicht.

Beispiel 2: Statisch finit, dynamisch finit, terminiert nach 11 Schritten.

3. Euklidischer Algorithmus:

(Voraussetzung/Festlegung: A ist anfangs ungleich B, ansonsten ist der GGT bereits bekannt.)

```
A := 100;
B := 35;

REPEAT
  IF B > A THEN
    BEGIN
      T := B;
      B := A;
      A := T;
    END;
  A := A - B;
UNTIL A = B;

GGT := B;
```

Alternativ:

```
A := 100;
B := 35;

REPEAT
  IF B > A THEN
    B := B - A;
  ELSE
    A := A - B;
  UNTIL A = B;

GGT := B;
```

4. Der Algorithmus ist so nicht korrekt, da sich bei drei Eingabewerten der Mittelwert aus $\frac{\text{Summe Eingabewerte}}{3}$ berechnet, und das Ergebnis bei Teilen stattdessen durch 4 falsch wird. In der Zeile `mittel := mittel + a + b + c;` ist zwar von der Logik her das `mittel` auf der rechten Seite zuviel, aber da es am Anfang auf 0 gesetzt wird, stört es an dieser Stelle nicht weiter. In der letzten Zeile muss also die Zahl 4 durch eine 3 ersetzt werden, damit das Ergebnis richtig wird.

5. Exakt nach Aufgabenstellung, die Werte 10 für a und 100 für n sind hier nur Beispiele:

```
a := 10;  
n := 100;  
sum := 0;  
FOR i := 1 TO n DO  
  sum := sum + a;
```

Komplexitätsklasse: $O(N)$ (Anzahl der Schritte/Instruktionen steigt linear mit n).

6. Effizienter:

```
a := 10;  
n := 100;  
sum := a * n;
```

Komplexitätsklasse: $O(1)$, da die Anzahl der Schritte/Instruktionen nicht mehr von n abhängt.