Lösung zu Aufgabe 23

Zur Berechnung der Zeitkomplexität stellen wir eine Tabelle auf, in der für jede Zeile die Kosten einer Ausführung, die Anzahl Ausführungen und die sich damit ergebenden Gesamtkosten stehen.

Zeile	Pseudocode-anweisung	Kosten pro Ausführung	Anzahl Ausführungen	Gesamt-kosten
1	`z := 0;`	1	1	1
2	`i := 0;`	1	1	1
3	`while (i < n) do`	1	$n+1$	$n+1$
4	`  b[i] := (i+1) *a[i+1];`	6	n	$6n$
5	`  i    := i+1;`	2	n	$2n$
6	`od`			
7	`i := n-1;`	2	1	2
8	`y := b[n-1];`	3	1	3
9	`while (i >= 1) do`	1	n	n
10	`  y := y *x +b[i-1];`	5	$n-1$	$5n-5$
11	`  if (y = 0)`	1	$n-1$	$n-1$
12	`  then z    := z+1;`	2	$n-1$	$2n-2$
13	`       a[i] := 0;`	2	$n-1$	$2n-2$
14	`  else a[i] := 1;`	2	0	0
15	`  fi`			
16	`  i := i-1;`	2	$n-1$	$2n-2$
17	`od`			
	Summe:			$22n-4$

Die Berechnung der Kosten einer Ausführung zeigen wir an drei Beispielen, nämlich Zeile 4, Zeile 9 und Zeile 10. In Zeile 4 wird auf ein Feld des Vektors `a` zugegriffen, was Kosten von einer Einheit verursacht. Der Index muß jedoch als `i+1` berechnet werden, was noch einmal eine Kosteneinheit mit sich bringt. Der Ausdruck `a[i+1]` bedeutet also Kosten von 2 Einheiten. Der Wert dieses Feldes wird mit `i+1` multipliziert. Dies bringt Kosten von 2 Einheiten mit sich: eine Kosteneinheit für das Erhöhen von `i` um 1 und eine Kosteneinheit für die Multiplikation. Daß `i+1` schon für den Vektorzugriff berechnet wurde, spielt keine Rolle. Die rechte Seite der Zuweisung hat also die Kosten 4 Einheiten. Auf der linken Seite wird auf ein Feld des Vektors `b` zugegriffen, was wieder eine Kosteneinheit bedeutet. Die Zuweisung schließlich bringt noch einmal eine Kosteneinheit. Insgesamt erhält man Kosten von 6 Einheiten.

In Zeile 9 wird ein einfacher Vergleich durchgeführt. Dies bringt Kosten von einer Einheit. Daß der Vergleich dabei „zusammengesetzt“ ist (Vergleich auf größer *und* auf gleich), spielt keine Rolle.

In Zeile 10 wird auf den Vector `b` zugegriffen, was Kosten von einer Einheit verursacht. Wieder muß der Index berechnet werden, wodurch Kosten von einer weiteren Einheit entstehen. Die Berechnung des Produktes von `x` und `y` kostet ebenfalls eine Einheit und die Addition des Feldwertes von `b` eine weitere. Die rechte Seite hat also die Kosten 4 Einheiten (wie in Zeile 4). Im Gegensatz zur Zeile 4 steht aber in Zeile 10 kein Vektorzugriff auf der linken Seite der Zuweisung. Es fällt daher nur eine Kosteneinheit für die Zuweisung an. Zeile 10 hat damit insgesamt die Kosten 5 Einheiten.

Bei der Bestimmung der Ausführungszahlen ist zu beachten, daß die Bedingung einer `while`-Schleife immer genau einmal öfter ausgeführt wird als der Schleifenrumpf. Denn der Rumpf wird jedesmal ausgeführt, wenn die Bedingung wahr ist. Die Bedingung wird aber im Abbruchfall (also wenn sie falsch wird) noch einmal geprüft, dann wird jedoch der Rumpf nicht ausgeführt.

Da die Zeitkomplexität im schlechstesten Fall betrachtet werden sollte, wird nur der `then`-Teil der `if`-Anweisung in Zeile 11 berücksichtigt. Denn der `then`-Teil hat die Kosten 4 Einheiten, der `else`-Teil die Kosten 2 Einheiten. Der schlechteste Fall tritt also dann ein, wenn stets der `then`-Teil ausgeführt wird (was auch auftreten kann, z.B. wenn alle Felder des Vektors `a` den Wert 0 enthalten).

Christian Borgelt