Übungen zur Algorithmischen Bioinformatik I Blatt 6

Lisanne Friedrich

Mai 2021

1. Aufgabe (10 Punkte): Türme von Hanoi

Programmieren Sie sowohl eine rekursive als auch eine iterative Version der Türme von Hanoi. Das Programm muss die Anzahl n der verwendeten Scheiben, die Startposition und die Zielposition als Eingabeparameter akzeptieren. Nach dem Aufruf soll das Programm beide Algorithmen ausführen und die Züge ausgeben. Begründen Sie die Korrektheit der beiden Verfahren. Analysieren Sie das Laufzeitverhalten Ihrer Algorithmen und messen Sie die Laufzeit der Implementierungen jeweils in Abhängigkeit von der Anzahl n der verwendeten Scheiben
Die Anzahl der Züge, die man zum lösen braucht lautet $2^n - 1$.

iterative Version:

```
iterativeTürme(S,H,Z, n))
begin
    if n mod 2 ==0 then
        Richtung = "im"
    if n mod 2 ==1 then
        Richtung = "gegen den"
    while höhe(Z) ≠ n do
        verschiebe Scheibe n Richtung Uhrezigersinn;
        if Scheibe ≠ n verschiebbar then
            Verschiebe Scheibe ≠ n
end
```

Zum besseren Verständnis stellen wir uns hier die Stäbe im Kreis angeordnet vor.Die Startposition ist oben, dei Hilfsposition rechts und die Zielposition links.
Wenn eine gerade Anzahl an Scheiben vorhanden ist, so wird die kleinste Scheibe im, ansonsten gegen den Uhrzeigersinn verschoben. Wenn eine Scheibe verschoben werden kann, bei der es sich nicht um die kleinste handelt so wird das getan. Danach wird die Schleife neu aufgerufen.
Die Unterscheidung zwischen im und gegen den Uhrzeigersinn kann man weglassen, wenn das Ziel nur die Umlagerung auf einen Belibigen anderen Stab gewünscht ist.
Die Korrektheitsanalyse ist nicht trivial, vgl. mit
http://gki.informatik.uni-freiburg.de/teaching/ws1011/infoI/infoI20-handout4.pdf den Folien 11 bis 27.

rekursive Version:

```
rekursiveTürme(S,H,Z, n))
begin
    if (n > 0) then
        rekursiveTürme(S, Z, H, n-1);
        bewege oberste Scheibe von S nach Z;
        rekursiveTürme(H, S, Z, n-1)
end
```

Hier steht n für die Anzahl der Scheiben, S für die Startposition, H für die Hilfsposition und Z für die Zielposition.
Korrektheitsbeweis: Der Algorithmus ist terminierend da irgendwann der Aufruf mit n = 1 erfolgt. Die Korrektheit kann mit einem Induktionsbeweis bewiesen werden.
Induktionsanfang: n=1 $\Longrightarrow$ Scheibe wird direkt auf Zielposition gelegt.
Induktionsbehauptung: Der Algorithmus funktioniert für n Scheiben mit $n > 1$.
Induktionsbeweis:
Der Algorithmus bewegt zunächst n Scheiben von der Startposition zum Arbeitsbereich (funktioniert nach Induktionsbehauptung),
-bewegt dann die n+1-te Scheibe zur Hilfsposition (funktioniert nach Induktionsanfang)
-bewegt schließlich die n Scheiben von der Hilfsposition zur Zielposition (funktioniert nach Induktionsbehauptung).