Karlstads Universitet
Institutionen för matematik och datavetenskap

Teoretisk Datalogi
DVGA17

Komplexitetsteori

Skriven av:
Alexander Florean
florean.alexander@gmail.com
Emanuel Svensson
emansven100@student.kau.se

Handledare:
Kerstin Andersson

8 januari 2020

Innehåll

1 Inledning

1.1 Antaganden

2 Del 1: Turingmaskin

Inledning Uppgiften går ut på att skapa och analyzera en turingmaskin som efter antal a:n i input lägger till inversen av en sträng med 0:or och 1:or till höger, startandes med 0 vilket ger 0 -> 01 -> 0110 osv.

Beskrivning av maskinen

```
1. gå höger tills \_, ersätt \_ med 0
    1.1 gå vänster tills \_, gå 3

2. gå vänster tills \_
3. gå ett steg höger, om ej a, avsluta
4. om a, ersätt a med \_
    4.1 gå höger tills 0 eller 1, om \_ gå 4.2
        4.1.1a om 1, ersätt med y
           4.1.1a.1 gå höger tills \_, ersätt med x, gå 4.1.2
        4.1.1b om 0, ersätt med x
        4.1.1b.1 gå höger tills \_, ersätt med y, gå 4.1.2

        4.1.2 gå vänster tills \_ eller a, gå 4.1

    4.2 gå vänster tills \_ eller a, ersätt x med 0 och y med 1 på vägen
    4.3 gå 2
```

3 Del 2: Tornen i Hanoi

Inledning Vi har tre pinnar med givna respektive namn A, B, C, samt notationen N för antal ringar.
Regler:

1. Flytta en ring åt gången.
2. Lägg aldrig en större ring ovanpå en mindre.

Given rekursiv algoritm:

1. Flytta temporärt högen med N 1 ringar från pinne A till pinne B.
2. Flytta den största ringen (den enda ringen kvar på pinne A) till pinne C.
3. Flytta N 1 ringar från pinne B till pinne C.

Omskrivning görs för att komma fram till en tidskomplexitet av den rekursiva algoritmen till en rekursiv psudokod/funktion. Som kan beskrivas, funktionen H(antal ringar, som flyttas från pinne x, till pinne y)

Tidsfunktion

H(N, A, C)
H(N - 1, A, B) // kallar på sig självt
flytt(A,C) // flyttar ringen ett steg
H(N - 1, B, C) // kallar på sig självt

Med tiden i åtanke kan flytt(A,C) ses som en konstant ökning i tidskomplexiteten.
Därav kan ovan skrivas om till tidsfunktionen

$$\begin{aligned} T(N) &= T(N-1) + 1 + T(N-1) \\ &= 2T(N-1) + 1 \end{aligned} \qquad 2005/06/28 ver: 1.3 subfigpack$$

Med hjälp av inducering fås.

$$\begin{aligned} T(N) &= 2[2T(N-1-1) + 1] + 1 \\ &= 2^2 T(N-2) + 3 \\ &= 2^2 [2T(N-3) + 1] + 3 \\ &= 2^3 T(N-3) + 7 \end{aligned} \qquad 2005/06/28 ver: 1.3 subfigpack$$

Då k är antal ringar för funktionen $T(N) = 2^k T(N-k) + 2^k - 1$
När $k \rightarrow N$ så fås tidsfunktionen

$$\begin{aligned} T(N) &= 2^N T(N-N) + 2^N - 1 \\ &= 2^N T(0) + 2^N - 1 \\ &= 0 + 2^N - 1 = 2^N - 1 \end{aligned} \qquad 2005/06/28 ver: 1.3 subfigpack$$

Detta ger Tornen i Hanoi tidskomplexiteten $\mathcal{O}(2^n)$ då konstanten kan bortses.

4 Sammanfattning

Referenser

[1] Michael Sipser, *Introduction to the Theory of Computation Third Edition*, June 2012, Cengage Learning