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

7 januari 2020

Innehåll

1 Inledning

1.1 Antaganden

2 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

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

Med hjälp av inducering fås.

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

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

$$\begin{aligned} T(N) &= 2^N T(N-N) + 2^N - 1 \\ &= 2^N T(0) + 2^N - 1 \\ &= 0 + 2^N - 1 = 2^N - 1 \end{aligned}$$

$2005/06/28 ver : 1.3 subfigpack$

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

3 Sammanfattning

Referenser

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