Exam subject outline - Nicklas Jacobsen qmr656

Divide: Split problemet i mindre problmer
Conquer: Når problemerne er små nok, så løs dem på en triviel måde
Combine: Kombinerer løsningerne til en stor løsning på det samlede problem

Substitutions metode

Metoden består af 2 step:
1: Gæt en løsning 2: Matematisk induktions til bevis at løsningen er rigtig.

$$T(n) = 2T(n/2) + n$$

Vi gætter $T(n) = O(n \cdot lg(n))$ Vi substituerer ind:

$$\begin{aligned} T(n) &\leq 2(c(n/2)lg(n/2)) + n \\ &\leq cn \cdot lg(n/2) + n \\ &= cn \cdot lg(n) - cn \cdot lg(2) + n \\ &= cn \cdot lg(n) - cn + n \\ &\leq cn \cdot lg(n) \end{aligned}$$

Bemærk det kun gælder for $n > 1$

Master method

Hvis vi har en recurrence på følgende form:

$$T(n) = aT(n/b) + f(n)$$

Så har har $T(n)$ følgende asymptotiske grænser.

1: Hvis $f(n) = O(n^{log_b a - \epsilon})$ ved en konstant $\epsilon > 0$, så er det ensbetydende med at $T(n) = \Theta(n^{log_b a})$

2: Hvis $f(n) = \Theta(n^{log_b a})$, så $T(n) = \Theta(n^{log_b a} lg(n))$

3: hvis $f(n) = \Omega(n^{log_b a + \epsilon})$ for en konstant $\epsilon > 0$, og hvis $a \cdot f(n/b) \leq c \cdot f(n)$ for en konstant $c < 1$, så $T(n) = \Omega(f(n))$

Eksempel - Merge-sort

Merge-sort er en Divide-and-Conquer sorterings algoritme og foregår i $O(n \cdot lg(n)$ tid. Algoritmen deler listen af tal op rekursivt til mindre del-lister indtil del-listerne har ét element, og derved bliver anset som været sorteret. For hver del-liste samler (merger) algoritmen listerne til en større del-liste, ved linært at samligne elementerne i listerne. Når der kun er en del-liste tilbage, er listen sorteret.

Merge-sort kan deles op i to del-algoritmer: 1. Merge som er $O(n)$ og 2. Divide som er $O(lg(n)$. Og den rekursive form er derved $T(n) = 2(n/2) + n$.