Intro: Divide and conquer er en tilgang, hvor et problem bliver lavet om til flere subproblemer som ligner det originale problem men er mindre i størrelsen, hvorefter de slttes sammen for at lave en løsning til det originale problem.

Eksempel med merge sort: Hvad gør den (merge (theta(n)) comparer dem og merger, mergesort deler den op) - divide i n/2 - conquer, sorterer rekursivt - combine de 2 delproblemer.

Vis køretid på *rekusions ligning (T(n)* (s. 35+67 (aT(n/b) + D(n) + C(n)) løs med rekursions-træ, substitution og master method.

Rekursions-træ: Det træ der er tegnet, add køretid for hvert level (cn) og gæt på O(lg n * n). Dette giver kun en ide og ikke noget konkret.

Substitution The substitution method for solving recurrences consists of two steps:

- Guess the form of the solution.
- Use mathematical induction to find constants in the form and show that the solution works.

The inductive hypothesis is applied to smaller values, similar like recursive calls bring us closer to the base case

Example
The recurrence relation for the cost of a divide-and-conquer method is $T(n) = 2T(\lfloor n/2 \rfloor) + n$. Our induction hypothesis of $T(n)$ is $O(n \lg n)$ or $T(n) \leq cn \lg n$ for some constant c, independent of n.

Assume the hypothesis holds for all $m < n$ and substitute m = n/2 rundet ned:

$$
\begin{aligned}
T(n) &\leq 2(c\lfloor n/2 \rfloor \lg_2(\lfloor n/2 \rfloor)) + n && (1)\\
&\leq cn \lg_2(n/2) + n && (2)\\
&= cn \lg_2(n) - cn \lg_2(2) + n && (3)\\
&= cn \lg_2(n) - cn + n && (4)\\
&\leq cn \lg_2(n) && (5)
\end{aligned}
$$

as long as $c \geq 2$ (vises via basisstep).

Master method

Example

Consider the recurrence $T(n) = 2T(n/2) + f(n)$ of merge sort. We identify the variables values as $a = 2$ since each recursive call creates two subproblems and $b = 2$ since each of these are of $n/2$ the size of the previous problem. The combine step is of linear time $\Theta(n)$.

We then have that $O(n^{\lg_2 2}) = f(n)$, since $f(n) = \Theta(n)$ and this satisfies the previous equation, because $O(n^{\lg_2 2}) = \Theta(n)$. Therefore, the second case of the master theorem applies to this algorithm, and the complexity of it is then $\Theta(n^{lg_2 2} \lg n) = \Theta(n \lg n)$.

Quicksort

Hvad gør den (divide, conquer, combine (s 170)).

Proof ved loop invariant (s. 173).

evt: Worst case er $\theta(n^2)$. Best case er $\theta(nlgn)$. Meget tættere på best case end worst case

randomized quicksort.