Nogle formler fra DM507

$$n^a = o(b^n)$$

Eksempel:

$$n^{100} = o(1.25^n)$$

Pointe: Ethvert polynomium vokser asymptotisk langsommere end enhver eksponentialfunktion.

$$(\log_c n)^a = o(n^d)$$

Eksempel:

$$(\log_3 n)^{100} = o(n^{0.25})$$

Pointe: Enhver logaritme (selv opløftet i enhver potens) vokser asymptotisk langsommere end ethvert polynomium. [Bemærk at $(\log n)^a$ ofte skrives $\log^a n$.]

$$\log_a x = \frac{\log_b x}{\log_b a} = \Theta(\log_b x)$$

Eksempel:

$$\log_3 7 = \frac{\ln 7}{\ln 3} = \frac{1.9459\ldots}{1.0986\ldots} = 1.7712\ldots$$

Pointe: Alle logaritmer er skaleringer af hinanden. Formlen kan også bruges til at beregne alle logaritmer (dvs. alle grundtal) ud fra én logaritme (f.eks. ln) på lommeregner.

$$a^{\log_b n} = n^{\log_b a}$$

Eksempel:

$$4^{\log_2 n} = n^{\log_2 4} = n^2$$

Pointe: $\exp(\log(n))$ giver et polynomium.

$$\sum_{i=1}^{k} i = \frac{(k+1)k}{2} = \Theta(k^2)$$

Eksempel:

$$1 + 2 + 3 + 4 + \cdots + n = \Theta(n^2)$$

Pointe: Dette viser bla. at InsertionSort tager $\Theta(n^2)$ tid.

$$\sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha} = \Theta(1) \text{ for } |\alpha| < 1$$

Eksempler ($\alpha = 1/2$):

$$1 + 1/2 + 1/4 + 1/8 + \cdots = 2$$

$$\begin{aligned} & n + n/2 + n/4 + n/8 + \cdots \\ & \quad = n(1 + 1/2 + 1/4 + 1/8 + \cdots) \\ & \quad = 2n \end{aligned}$$

Pointe: I en exponentielt faldende udvikling dominerer første led det hele.

$$\sum_{i=0}^{k} \alpha^i = \frac{\alpha^{k+1} - 1}{\alpha - 1} = \Theta(\alpha^k)$$

Eksempel ($\alpha = 2$):

$$1 + 2 + 4 + 8 + 16 + 32 = 64 - 1 = 2 \cdot 32 - 1$$

Pointe: I en eksponentielt stigende udvikling dominerer sidste led det hele.

$$\sum_{i=1}^{k} \log i = \Theta(k \log k)$$

Eksempel:

$$\log 1 + \log 2 + \log 3 + \cdots + \log n = \Theta(n \log n)$$

Pointe: Dette viser f.eks. at TreeSort (gentagen indsættelse i et balanceret søgetræ) og HeapSort tager $\Theta(n \log n)$ tid.

$$\sum_{i=1}^{\infty} i\alpha^i = \frac{\alpha}{(1-\alpha)^2} = \Theta(1) \text{ for } |\alpha| < 1$$

Eksempel ($\alpha = 1/2$):

$$\begin{aligned}
&1 \cdot n/2 + 2 \cdot n/2^2 + 3 \cdot n/2^3 + \cdots + \log n \cdot 1 \\
&= n(1/2 + 2(1/2)^2 + 3(1/2)^3 + \cdots + \log n \cdot (1/2)^{\log n}) \\
&= \Theta(n)
\end{aligned}$$

Pointe: Dette viser, at en heap kan bygges i $O(n)$ tid.