a)

f(n) sei $\sum_{i=1}^{n-1}(n-i)$, so ist das $(n-1)+(n-2)+...+(n-(n-1))$ in umgekehrter Reihenfolge also $1+2+3+...+(n-2)+(n-1)$ und das ergibt mit "dem kleinen Gauß": $\sum_{i=1}^{n-1}(i) = \frac{n(n-1)}{2} = \frac{n^2-n}{2}$

Nun zur asymptotischen Betrachtung:

sei f(n) wie oben gegeben und g(n) = n^2, dann gilt zu zeigen:

$f = O(g)$ und $f = \Omega(g)$

$\underline{f = O(g)}$: mit c = 1 lässt sich leicht erkennen: ab $n_0 = 2$ gilt:

$f(n) \leq c \cdot g(n)$, denn $\frac{n^2-n}{2} < n^2$

$\underline{f = \Omega(g)}$: mit c = 4 lässt sich leicht erkennen: ab $n_0 = 2$ gilt:

$g \leq c \cdot f(n)$, bzw. $c \cdot f(n) \geq g(n)$ denn

$4 \cdot \frac{n^2-n}{2} = 2n^2 - 2n = n^2 + (n^2 - 2n) \geq n^2$, ($n^2 \geq 2n$ gilt ab n =2)

$\underline{\text{Damit gilt: } f(n) = \sum_{i=1}^{n-1}(n-i) = \Theta(n^2)}$

b)

Probieren wir mal was mit n^n. Ist $(n+1)^{n+1} = \Theta(n^n)$?

Das würde bedeuten es gäbe eine Konstante c, sodass $(n+1)^{n+1} \leq c \cdot n^n$

$(n+1)^{n+1} = (n+1)^n \cdot (n+1) \geq n^n \cdot n$ und damit müsste c größer als n sein, für alle $n > n_0$, so ein c gibt es nicht, damit folgt:

Es gibt Funktionen für die nicht gilt $f(n+1) = \Theta(f(n))$

c)

Sei $f(n) = O(n^r) \Rightarrow \exists c_k : f(n) \leq c_k \cdot n^r$ ab $n > n_0$

Was ist mit $log(f(n))$? Da der Logarithmus monoton wachsend auf $\mathbb{R}_{\geq 0}$ ist gilt damit $log(f(n)) \leq log(c_k \cdot n^r)$ und mit den logarithmusgesetzen folgt: $log(f(n)) \leq log(c_k) + r \cdot log(n)$, und damit gilt: Es gibt eine Konstante c_r, sodass $log(f(n)) \leq c_r \cdot log(n)$

d)

Untersuchen wir den Bruch $\frac{f(n)}{n^r}$ so gilt mit $lim_{n\to\infty} fracf(n)n^r = 0$,der Argumentation aus c) und den Logarithmusgesetzen folgt:
$lim_{n\to\infty}\frac{log(f(n))}{log(n^r)} = 0$ und somit $\frac{1}{r} lim_{n\to\infty}\frac{log(f(n))}{log(n)} = 0$.
Also $log(f(n)) = o(log\ n)$

e)

$f_1 = n^{1,5}$; $f_2 = log(n)$; $f_3 = nlog(n)$; $f_4 = n$; $f_5 = O(1)$
Assymptotische Reihenfolge: $f_5 < f_2 < f_4 < f_3 < f_1$

Mehr?

Prüfen wir $n^{log_b(a)}$

a)

$(n^{log_3(27)}) = n^3 = \Omega(n^9)$; $\Theta(n^9)$

b)

$2 + 3 \cdot (T(n-3)) = 2 + 3 \cdot (2 + 3T(n-6)) = 2 \cdot 3^0 + 2 \cdot 3^1 + 3^2(2 + T(n-9))$
$= \sum_{i=0}^{i=n/3} 2 \cdot 3^i = 2\frac{3^{1+n/3}-1}{2} = 3 \cdot 3^{n/3} - 1 = \Theta(3^{n/3})$

c)

$n^{log_3(1)}$, also $log(n) = \Omega(n^{0+\epsilon})$, das gilt nicht!
Es ist auch nicht $log(n) = \Theta(n)$ oder $(log\ n) = O(n^{-\epsilon})$
Also kein Master-Theorem!
Die Untersuchung ergibt bei T(0) begonnen:
$T(1) = 0$; $T(3) = T(1){+}1$; $T(9) = (T(3){+}2) = ((T(1){+}1){+}2) = 0{+}1{+}2$
$T(n) = \sum_{i=0}^{log_3(n)} i = \frac{\log_3(n)(log_3(n)+1)}{2} = (k^2 + k)/2$, mit $k = log_3(n)$

mal mit Algo

Wir laufen immer n mal eine kleine for schleife durch. In jedem Durchlauf ändert sich der Endwert der kleinen Schleife.

a)

n-1 Durchläufe + n-2 + n-3 +...+ 0 Durchläufe, wenn i = n, also

$0 + 1 + 2 + ... + n - 2 + n - 1 = \sum_{i=0}^{n-1} = \frac{n(n-1)}{2}$

b)

log(1) Durchläufe + log(2) Durchläufe + log(3) + log(4)+...+log(n-1)+log(n)

Mit den Logarithmusgesetzen folgt:

$log(1 \cdot 2 \cdot 3 \cdot ... \cdot (n-1) \cdot n) = log(n!) = \Theta(nlog(n))$

c

$\sqrt{1} + \sqrt{2} + ... + \sqrt{n} = \sum_{i=1}^{n} i^{1/2} = \Theta(n^{1,5})$

Wälder und Bäume?

pfff..... kommt noch. vielleicht...