Antworten zu Übungsblatt Nr. 2

Aufgabe 1

ZZ: $\phi = y/x$ ist Lösung für $z^2 = z + 1$.

$$x/y = y/(x+y) <=> y/x = (x+y)/y <=> \phi = \phi^{-1} + 1 <=> \phi^2 = 1 + \phi$$

ZZ: Zweite Lösung für Quadratische Gleichung $z^2 = z + 1$.

$$x/y = y/(x+y) <=> (x+y)/y = y/x <=> x/y + 1 = y/x <=> -y/x + 1 = -x/y$$
$$<=> +xy/xy + (-x/y) = (-x/y)^2 <=> +xy/xy + (-x/y) = (-x/y)^2$$
$$<=> 1 - x/y = (-x/y)^2 <= \psi = -x/y => \psi^2 = \psi + 1$$

Lösung 1: $\phi = y/x$, Lösung 2: $\psi = -x/y$.

Aufgabe 2

ZZ: $F_n =$ Fibonacci-Zahl der stelle n.
IA: $F_0 = (\phi^0 - \psi^0)/\sqrt{5} = (1-1)/\sqrt{5} = 0$
IS: $F_{n+1} = F_n + F_{n-1}$.
ZZ: $\phi^n = \phi^{n-1} + \phi^{n-2}$, sowie $\psi^n = \psi^{n-1} + \psi^{n-2}$.
Bew.: Sind beide dadurch zu erreichen, in dem man die Gleichung $z^2 = z + 1$ mit z^{n-2} multipliziert, sowie (wie bereits bewiesen) ϕ oder respektive ψ einsetzt.

$F_{n+1} = (\phi^{n+1} - \psi^{n+1})/\sqrt{5} = (\phi^n + \phi^{n-1} - \psi^n - \psi^{n-1})/\sqrt{5} = (\phi^n - \psi^n)/\sqrt{5} + (\phi^{n-1} - \psi^{n-1})/\sqrt{5} = F_n + F_{n-1}$.

Aufgabe 3

$T(n) \leq A * n + 3 * T(n/2)$
$T(n) \leq A * n + 3 * (A * n/2 + 3 * T(n/4))$
$T(n) \leq A * n + 3 * A * n/2 + 9 * T(n/4)$
$T(n) \leq A * n + 3 * A * n/2 + 9 * (A * n/4 + 3 * T(n/8))$
$T(n) \leq A * n + 3 * A * n/2 + 9 * A * n/4 + 27 * T(n/8)$
...
$T(n) \leq A * n * \sum_{i=0}^{k-1} (3/2)^i + 3^k * T(n/(2^k))$

mit $k = log_2 n$ folgt dadurch: $3^k = 3^{log_2 n} = (2^{log_2 3})^{log_2 n} = (2^{log_2 n})^{log_2 3} = n^{log_2 3}$
$= n^{log_2 3} * T(1) + \sum_{i=0}^{log_2 n-1} (3/2)^i * n * A = n^{log_2 3} * A + \sum_{i=0}^{log_2 n-1} (3/2)^i * n * A$
$\sum_{i=0}^{log_2 n-1} (3/2)^i = (1 - (3/2)^{log_2 n})/(-0.5) = -2 + 2 * (n^{log_2 3)/n}$
$= 3 * n^{log_2 3} * A - 2 * A * n \leq 3 * n^{log_2 3} * A$. mit $A' = 3 * A$ folgt: $= \ A' * n^{log_2 3}$

Aufgabe 4

Gegenbeispiel dafür, dass immer mindestens $v \geq log_2(n!)$ Vergleiche Passieren wäre einfach TimSort. TimSort ist eine Kombination aus MergeSort und InsertionSort, welche zwar beide im Worst case $v \geq log_2(n!)$ Vergleiche anstellen, aber nicht im Best Case, welcher damit ein explizites Gegenbeispiel darstellt. Ist die Liste also bereits sortiert werden genau $v = n - 1$ vergleiche durchgeführt, und der Algorithmus danach Beendet. v ist hier also offensich kleiner als $log_2(n!)$.

"It has no bad cases (O(N log N) is worst case; N-1 compares is best" - https://mail.python.org/pipermail/py dev/2002-July/026837.html