Aufgabe 3

a)

$$\begin{array}{c|c|l}
* & \begin{matrix}0 & 1\\ 1 & 1\end{matrix} & = \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^1 = \begin{pmatrix}F_0 & F_1\\ F_1 & F_2\end{pmatrix} \\ \hline
\begin{matrix}0 & 1\\ 1 & 1\end{matrix} & \begin{matrix}1 & 1\\ 1 & 2\end{matrix} & = \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^2 = \begin{pmatrix}F_1 & F_2\\ F_2 & F_3\end{pmatrix} \\ \hline
\begin{matrix}0 & 1\\ 1 & 1\end{matrix} & \begin{matrix}1 & 2\\ 2 & 3\end{matrix} & = \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^3 = \begin{pmatrix}F_2 & F_3\\ F_3 & F_4\end{pmatrix} \\ \hline
\begin{matrix}0 & 1\\ 1 & 1\end{matrix} & \begin{matrix}2 & 3\\ 3 & 5\end{matrix} & = \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^4 = \begin{pmatrix}F_3 & F_4\\ F_4 & F_5\end{pmatrix}
\end{array}$$

Annahme:

$$\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^n = \begin{pmatrix}F_{n-1} & F_n\\ F_n & F_{n+1}\end{pmatrix}$$

Für n → 0: bereits bewiesen (s.o.).

Für n → n +1:

$$\begin{array}{c|c|l}
* & \begin{matrix}F_{n-1} & F_n\\ F_n & F_{n+1}\end{matrix} & \\ \hline
\begin{matrix}0 & 1\\ 1 & 1\end{matrix} & \begin{matrix}F_n & F_{n+1}\\ F_{n+1} & F_{n+2}\end{matrix} & = \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^{n+1}
\end{array}$$

Ergebnismatrize an der Stelle 2x2 $=\ F_{n+2}$ da $F_x + F_{x+1} = F_{x+2}$.

Da $F_0 = 0$ und $F_1 = 1$:

$$\begin{aligned}
& \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^n * \begin{pmatrix}F_0\\ F_1\end{pmatrix} \\
= & \begin{pmatrix}F_{n-1} & F_n\\ F_n & F_{n+1}\end{pmatrix} * \begin{pmatrix}0\\ 1\end{pmatrix} \\
= & \begin{pmatrix}F_n\\ F_{n+1}\end{pmatrix}
\end{aligned}$$

Da $\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^n$ immer $\begin{pmatrix}F_{n-1} & F_n\\ F_n & F_{n+1}\end{pmatrix}$ ist und die Multiplikation mit dem Vektor $\begin{pmatrix}0\\ 1\end{pmatrix}$ immer nur die zweite Spalte ausgibt ist das Ergebnis dieser Formel immer $\begin{pmatrix}F_n\\ F_{n+1}\end{pmatrix}$.

b)

Eine effizientere Berechnung von X^n lässt sich mit dem folgenden Pseudocode darstellen:

1: **function** $X^n_{(X, n)}$
2:     **if** ((X > 1) & (X % 2 = 0))
3:         **return** $(X^n_{(X, n/2)})^2$
4:     **else if** ((X > 1) & (X % 2 != 0))
5:         **return** $(X^n_{(X, (n-1)/2)})^2 * X$
6:     **else**
7:         **return** X
8: **end function**

Diese rekursive Funktion ruft sich immer wieder selber auf bis n = 1 ist. Dabei wird n in jedem Schritt halbiert (Zeile 2, 3). Sollte n nicht ganzzahlig halbierbar sein so wird n-1 halbiert und das fehlende X manuell hinzu multipliziert (Zeile 4, 5). Sobald n den Wert 1 erreicht hat wird X zurückgegeben und der Algorithmus terminiert (Zeile 6, 7, 8).

Da die Laufzeit von n abhängt und dieses sich superlinear verringert, ist die Laufzeit sublinear. Setzt man für n z.B. 32 ein, so wird dieser Algorithmus 6 mal durchlaufen (32, 16, 8, 4, 2, 1). 6 entspricht dabei ln(32). Der Algorithmus wird also ln(n) mal durchlaufen und seine Laufzeit lässt sich mit dem Term $X^n_{(X, n)} \in O(\log n)$ beschreiben.

c)

Das Matrizen-Verfahren lässt sich auch als Pseudocode ausdrücken:

function $MatFib_{(n)}$		
	e $\rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$	
	while (n > 1)	$(n-1)$
	e $\rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} * e$	$(n-1) * (8 * l^{1.58} + 4 * l)$
	n- -	
	return $e * \begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$1 * (4 * l^{1.58} + 2 * l)$
end function		

Diese iterative Berechnung der Fibonaccizahlen führt genau n Rechenoperationen durch (n-1 mal die while-Schleife und 1 mal zur Berechnung der Ausgabe). In der while-Schleife werden je Durchlauf 8 Multiplikationen und 4 Additionen durchgeführt ($8 * l^{1.58} + 4 * l$). Bei der Ausgabe werden einmalig 4 Multiplikationen und 2 Additionen durchgeführt ($4 * l^{1.58} + 2 * l$). Die Laufzeit lässt sich damit wie folgt berechnen:

$$(n-1) * (8 * l^{1.58} + 4 * l) + (4 * l^{1.58} + 2 * l)$$

Laut Vorlesung kann eine Fibonaccizahl bis zu n Bits beanspruchen. Setzt man n für l ein so ergibt sich (abzüglich aller Konstanten) für $MatFib_{(n)}$ ebenfalls eine quadratische Laufzeit. Zu zeigen, dass das Matrizen-Verfahren echt schneller läuft als das in der Vorlesung gezeigte Verfahren ist mir daher nicht möglich.