Algorithms example sheet 2

Exercise 22

```
def f(n):
    if n < 2:
        1
    else:
        f(n-2) + f(n-1)
```

Say $r(n)$ is the number of recursive calls to compute `f(n)`. Then, $r(n) = r(n-2) + r(n-1)$, where $r(0), r(1) = 1$, so in fact $r(n)$ is the same as `f(n)`.

Exercise 23

$$\begin{bmatrix}1\end{bmatrix} \times \begin{bmatrix}1 & 1\end{bmatrix} \times \begin{bmatrix}1\\1\end{bmatrix}$$

Multiplying the right matrices first gives $1 \times 1 + 1 \times 1$, and then 1×2, for a total of 3 multiplications.

Multiplying the left matrices first gives $1 \times 1,\ 1 \times 1$ and then $1 \times 1 + 1 \times 1$, for a total of 4 multiplications.

Exercise 33

$$\begin{gathered}
f(n) = f(n/2) + k \\
\text{Let } n = 2^m \text{, so } m = \log_2 n. \\
f(n) = f(2^m) = f(2^m/2) + k \\
= f(2^{m-1}) + k \\
= (f(2^{m-2}) + k) + k = f(2^{m-2}) + 2k \\
= (f(2^{m-3}) + k) + 2k = f(2^{m-3}) + 3k \\
\dots \\
= f(2^{m-m}) + mk \\
= f(1) + mk \\
\therefore f(n) = k_0 + k \log_2 n
\end{gathered}$$