Jeffrey
Lansford

2.23 Let A be an array with the elements $[1 \dots n]$, then

$A_1 = [1 \dots n/2]$

$A_2 = [n/2 \dots n]$

Then we find the majority for both of these halfs. If both parts have the same majority, then that element is the majority of A. If one part has majority element, then count both parts to see if it is really the majority element. Using the Master Theorem we have $2T(n/2) + O(n) = O(n \log n)$.

2.26 n^2 is equal $n * n$, thus it would take the same time to square a n-bit number and multiply two n-bit numbers. Thus one of them is ~~not~~ faster than the other. The Professor is wrong.

2.28

$$H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$$

The product of $H_k v$ can be written as

$$H_k v = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{matrix} H_{k-1}v_1 + H_{k-1}v_2 \\ H_{k-1}v_1 - H_{k-1}v_2 \end{matrix}$$

$$= \begin{matrix} H_{k-1}(v_1 + v_2) \\ H_{k-1}(v_1 - v_2) \end{matrix}$$

The runtime can be described as $T(n) = 2T(n/2) + O(n)$

which would equal $O(n \log n)$ by Master Theorem