2.19

A merge of two size m arrays takes $O(m \log m)$. So the first merge takes $k \log k$, the second $2k \log 2k$ and the nth $nk \log nk$. So the overall time taken is $\sum_i ik \log ik = k \sum_i i \log ik = knO(n \log nk) = O(n^2 k \log nk)$.

part b

Suppose we have n size k lists. We can merge them into $n/2$ size $2k$ lists in $O(nk \log k)$. We then merge these into $n/4$ lists etc. So $T(n,k) = T\left(\frac{n}{2}, 2k\right) + O(nk \log k)$. There are $\log n$ iterations here, and the ith iteration takes $ik \log 2^i k = i^2 k \log 2k = O\left((\log n)^2 k \log k\right)$ (since i can be at most $\log n$). So the whole thing takes $O\left((\log n)^3 k \log k\right)$.

2.23

T1: If x is not the majority element of at least one of A and B, then it cannot be the majority element of $A \cup B$. Proof: Divide A into those which are x and those which aren't. Say that the number which are x is a_x, call the number which are not x a_y and the same for B. We want to show that $a_x + b_x > a_y + b_y \implies (a_x > a_y) \vee (b_x > b_y)$. Suppose they were both less than their respective y's, yet the inequality held. Then we could subtract $(a_x + b_y)$ from both sides, and end up with $0 > n$ for some positive integer n, which is a contradiction. QED.

So our algorithm: divide S into A and B. Find the majority of A and B. Check if they're equal. If so, we're done. Otherwise, scan B to see if the majority of A is the majority of $A \cup B$ and vice versa. This scanning will take linear time.

So $T(n) = 2T(n/2) + O(n)$ which is $O(n \log n)$ by the master theorem.

Part b

To prove the first property, just note that you either throw out both or one of each pair, so the most you could end up with is $n/2$.

For the second: suppose x is a majority of A. Then it must be paired with itself at least once (by the pidgeonhole principle). Furthermore, any time another element is paired with itself, then those elements are "taken away" from a pairing with x, so x gets paired with itself again. So the majority will always remain the majority.

For example, if we had (x,y), (x,y), then in order to pair (y,y) we'd also need to pair (x,x).

So this is $T(n) = T(n/2) + O(n) = O(n)$ by the master theorem.