The following illustration of linear time is not from the book, so I am including these notes here. Suppose we have a K-bit binary counter that we increment from the all 0's situation. Below is part of the process with K = 4.

0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
⋮
1 1 1 1

Note that in each transition, we scan the counter from right to left, ~~flipping~~ till we find a 0. We then flip all bits we have scanned.

So, in the transition $0011 \rightarrow 0100$, we flip 3 bits.

We want to upper bound the <u>overall numbers of flips</u> over all the transitions.

Let $n = 2^k - 1$. The process steps ~~until~~ when we reach the all 1's situation. Since there are n transitions, and each involves at most k ~~bits~~ bit flips, the total number of flips is at most $n \times k$, which is $O(n \log n)$.

Can we give a better bound?

To do this, we categorize a bit flip as either a 0-to-1 flip or a 1-to-0 flip. Each transition has exactly one 0-to-1 flip, so the total number of 0-to-1 flips is n.

The number of 1-to-0 flips varies across transitions, and is therefore harder to count. However, we make the following simple but important observation: the number of 1-to-0 flips overall is bounded by the number of 0-to-1 flips. This is because, if you focus on the j^{th} bit of a counter, it starts at 0, and over time, alternates between 1 and 0. So we can account for any occasion it flips from 1 to 0 by charging to the previous 0 to 1 flip.

Thus, the overall number of 1-to-0 flips is bounded by the overall number of 0-to-1 flips, which is exactly n.

Thus, the overall number of bit flips is at most $2n$, which is $O(n)$.