DAM3 states the standard 2-color structure theorem and hints that there are nonconstructive proofs. But it doesn’t give any. Here we give two.

**Theorem.** $G$ is 2-colorable $\iff G$ has no odd cycles.

The easy part is $\implies$ and all proofs of this part are essentially the same: Assuming $G$ is two colored, as you go around any cycle you alternate colors, so you must take an even number of steps to get back to the start. So no cycle is odd.

The differences between constructive and nonconstructive proofs show up in the $\iff$ part, which we do now.

**Proof 1:** Assume all cycles of $G(V, E)$ are even. Let $C$ be a maximal 2-coloring, that is, a 2-coloring of as many vertices of $G$ as can be 2-colored properly. We must show that this maximal coloring colors all of $V$.

Suppose not. Then some $v$ is not colored. But it must be that two vertices $u$ and $u'$ adjacent to $v$ are colored, and colored different colors, or else we could color $v$ the other color, contradicting maximality.

Let $G'$ be the subgraph of $G$ consisting of all colored vertices and all edges of $G$ between pairs of them.

Now there are 2 cases: either $u$ and $u'$ are in the same connected component of $G'$ or they are not.

If they are, there is a path between them on which all vertices are colored. Since $u, u'$ have the same color, this path $u, \ldots, u'$ has even length. But then the cycle $v, u, \ldots, u', v$ using this path has odd length, a contradiction. So case 1 doesn’t happen.

In case 2, reverse the colors in the component of $u$. This component is still properly 2-colored, but now $u, u'$ are the same color and we are free to color $v$ the other color — again a contradiction to maximality. (Well, there may still be other vertices $u'', u''', \ldots$ adjacent to $v$ of the other color, but by the same argument we have just given, they all must be in different components and we can change their color one by one.)

In summary, it is contradictory to suppose that the whole of $G$ cannot be 2-colored.

**Proof 2:** In each connected component of $G$, define a coloring as follows. Pick any vertex $v$ in the component. Divide the other vertices of the component into two sets: those on a path of even length from $v$, and all the others. Color those in the first set the same color as $v$, those in the other set the other color.

We claim that this coloring is proper, and thus $G$ is 2-colorable. Suppose not. Then two adjacent vertices $u, u'$ are the same color. Because they are adjacent, they are in the same component. If they are the same color as $v$ for that component, they each must be on an even-length path to $v$. Then the cycle $v, \ldots, u, u', \ldots, v$ using these paths has odd length, a contradiction. If they are the other color, they each must be on an odd-length path to $v$. Then the cycle $v, \ldots, u, u', \ldots, v$ using these paths is again odd length, a contradiction.
Note 1: You shouldn’t find it too hard to turn Proof 2 into a constructive one. Proof 1 is harder, but can be done. But why bother? Best to build a constructive proof to begin with. And, as the text points out, if we are going to do that, then why not state the theorem constructively.

Note 2: The proofs above, especially Proof 1, may seem convoluted, but they were the style for a long time. They do avoid needing or using the details of an algorithm, which the approach in DAM3 cannot.

Note 3: Nowhere above did we limit ourselves to simple cycles. If fact

\[
every \text{ cycle in } G \text{ is odd } \iff \text{ every simple cycle in } G \text{ is odd.}
\]

but \(\iff\) is a little tricky to prove. Also, if instead we had made the original theorem be \(G\) is 2-colorable \(\iff\) \(G\) has no simple odd cycles, then the proof would get harder. For instance, even if \(u, \ldots, v\) and \(u', \ldots, v\) are simple paths, the cycle \(v, \ldots, u, u', \ldots, v\) could be far from a simple cycle.

Note 4: There is in fact a very general algorithmic version of the 2-coloring theorem, more general than the one in DAM3. Let \(A\) be any algorithm that attempts to properly 2-color graphs, with the only restriction being that it colors vertices one at a time and it always picks for its next vertex a vertex adjacent to some vertex already colored. Then \(G\) is 2-colorable iff \(A\) succeeds. In other words, you can’t really go wrong testing 2-colorability, no matter how naive your algorithm. If you algorithm fails to 2-color, it’s not because it wasn’t smart enough.