--- Part Two ---

The second check is slightly more complicated: you need to find the value of the root node (`A` in the example above).

The *value of a node* depends on whether it has child nodes.

If a node has *no child nodes*, its value is the sum of its metadata entries. So, the value of node `B` is `10+11+12=33`, and the value of node `D` is `99`.

However, if a node *does have child nodes*, the metadata entries become indexes which refer to those child nodes. A metadata entry of `1` refers to the first child node, `2` to the second, `3` to the third, and so on. The value of this node is the sum of the values of the child nodes referenced by the metadata entries. If a referenced child node does not exist, that reference is skipped. A child node can be referenced multiple time and counts each time it is referenced. A metadata entry of `0` does not refer to any child node.

For example, again using the above nodes:

- Node `C` has one metadata entry, `2`. Because node `C` has only one child node, `2` references a child node which does not exist, and so the value of node `C` is `0`.
- Node `A` has three metadata entries: `1`, `1`, and `2`. The `1` references node `A`'s first child node, `B`, and the `2` references node `A`'s second child node, `C`. Because node B has a value of `33` and node `C` has a value of `0`, the value of node `A` is `33+33+0=`*`66`*.

So, in this example, the value of the root node is `66`.

What is the value of the root node?

Your puzzle answer was `21810`.