(7) [15 points]

Your friend Brutus the Bruin is a little weirded out by putting an entire tree in an array when implementing Min-Heaps (or Max-Heaps). He prefers the comfort of his `Node*` class and explicitly building the links, because it avoids the index calculations. Because he isn't such a good programmer, either (maybe that's why he doesn't really understand Heaps?), he asks you to help with his project of implementing a Min-Heap in this way.

He would like your help in writing the function `removeMin`. Your function needs to be correct, but as you may have guessed, keeping the heap at height $O(\log n)$ is much harder, and Brutus is not expecting you to accomplish this.

```
struct Node {
  Node *left, *right, *parent;
  int priority; // smaller means higher priority.
  int element;
    // this is an index of where to find the actual element in an array somewhere
};

void removeMin (Node *root)
/* yours to implement. You don’t need to keep the heap balanced, but
of course, you do have to make sure you keep a heap.
If you want helper functions, you may add them. */
```