Make One Move, Evaluate, and Update

```
p[pnum] -= count;
```
Modify the array in place (rather than creating a copy in our stack frame).

Recurse.

```
value = -nim (p);
p[pnum] += count;
```
Restore the original array value.

```
if (max < value) {
    max = value;
}
```
If this move's value is better than any previous move, record it.

`nim` is On the Web Page

The code is on the web page.

A Few Other Applications of Recursion

Other applications of recursion include…
◦ puzzles, such as Sudoku,
◦ code generation, and
◦ code optimization.

Generally, recursion is useful for wide searches (many children).

Deep searches (many levels) tend to break the stack.

Time for Another Think-Pair-Share

As before, let's do a group exercise in lecture.

The process:
1. I give you a problem.
2. You form groups of 3-4 people.
3. Talk about ways to solve the problem.
4. Once enough of the groups have finished, one group volunteers to share their answer.
5. We go over the group's answer together.