Recursion

Computing

The two parts every recursive function needs, and when to avoid it.

Every correct recursive function has exactly these parts.

1. **The base case**
   The smallest input, answered directly with no further calls.

2. **The recursive case**
   Reduce the problem, call yourself, combine the result.

3. **Guaranteed progress**
   Each call must move measurably toward the base case.

FACTORIAL

```
if n <= 1: return 1
return n * factorial(n - 1)
Progress: n decreases every call
```

Each call costs a stack frame. Depth beyond a few thousand overflows; rewrite as a loop or add memoisation.