10.4
Recursion as self reductions

Recursion

Reduction: reduce one problem to another

Recursion: a special case of reduction

1. reduce problem to a smaller instance of itself
2. self-reduction

1. Problem instance of size n is reduced to one or more instances of size $n-1$ or less.
2. For termination, problem instances of small size are solved by some other method as base cases

Recursion

Reduction: reduce one problem to another

Recursion: a special case of reduction

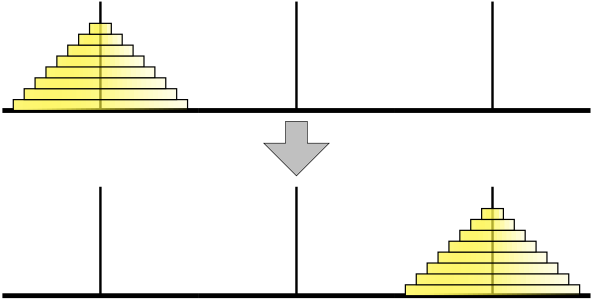
1. reduce problem to a smaller instance of itself
2. self-reduction

1. Problem instance of size n is reduced to one or more instances of size $n-1$ or less.
2. For termination, problem instances of small size are solved by some other method as base cases

Recursion

1. Recursion is a very powerful and fundamental technique
2. Basis for several other methods
    1. Divide and conquer
    2. Dynamic programming
    3. Enumeration and branch and bound etc
    4. Some classes of greedy algorithms
3. Makes proof of correctness easy (via induction)
4. Recurrences arise in analysis

Tower of Hanoi



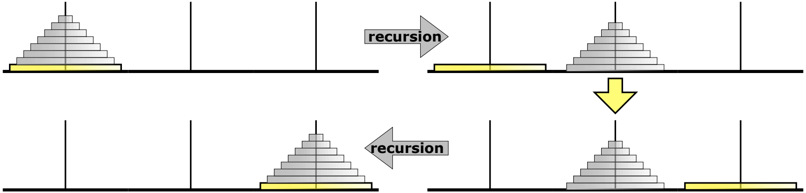
The Tower of Hanoi puzzle

Move stack of n disks from peg $\mathbf{0}$ to peg $\mathbf{2}$, one disk at a time.

Rule: cannot put a larger disk on a smaller disk.

Question: what is a strategy and how many moves does it take?

Tower of Hanoi via Recursion



The Tower of Hanoi algorithm; ignore everything but the bottom disk

Recursive Algorithm

```
Hanoi(n, src, dest, tmp):
    if (n > 0) then
        Hanoi(n − 1, src, tmp, dest)
        Move disk n from src to dest
        Hanoi(n − 1, tmp, dest, src)
```

$T(n)$: time to move n disks via recursive strategy

$$T(n) = 2T(n-1) + 1 \qquad n > 1 \qquad \text{and } T(1) = 1$$

Recursive Algorithm

```
Hanoi(n, src, dest, tmp):
    if (n > 0) then
        Hanoi(n − 1, src, tmp, dest)
        Move disk n from src to dest
        Hanoi(n − 1, tmp, dest, src)
```

$T(n)$: time to move n disks via recursive strategy

$$T(n) = 2T(n-1) + 1 \qquad n > 1 \qquad \text{and } T(1) = 1$$

Recursive Algorithm

```
Hanoi(n, src, dest, tmp):
    if (n > 0) then
        Hanoi(n − 1, src, tmp, dest)
        Move disk n from src to dest
        Hanoi(n − 1, tmp, dest, src)
```

$T(n)$: time to move n disks via recursive strategy

$$T(n) = 2T(n-1) + 1 \qquad n > 1 \qquad \text{and } T(1) = 1$$

Analysis

$$
\begin{aligned}
T(n) &= 2T(n-1)+1 \\
&= 2^2T(n-2)+2+1 \\
&= \ldots \\
&= 2^iT(n-i)+2^{i-1}+2^{i-2}+\ldots+1 \\
&= \ldots \\
&= 2^{n-1}T(1)+2^{n-2}+\ldots+1 \\
&= 2^{n-1}+2^{n-2}+\ldots+1 \\
&= (2^n-1)/(2-1)=2^n-1
\end{aligned}
$$

THE END

...

(for now)