Recursive Functions

Friday, March 13, 2015 3:08 PM

Recursive functions

recursive := a function which calls itself.

break a problem up into smaller/simpler vesions of the problem, solve those problems recursively, combine results to solve original problem.

Example

$n! = n(n-1)(n-2)\cdots 1 \qquad 0! = 1$

function factorial(n)

if $n = 0$, return 1 // base case

else, return n * factorial (n-1) // recursive step

$$n! = n\boxed{(n-1)!}$$

smaller version of the problem

iterative version

```
function factorial2(n)
    f = 1
    for i = 1 to n
        f = f * i
    return f // 1*2*3*...*n = n!
```

Example $a^n = a \cdot a^{n-1}$

function expon(a,n)

if n=0, return 1 //base case

else, return $a\,\cancel{*}\,\underbrace{\text{expon}(a,n-1)}_{a^{n-1}}$

fast exponentiation

$$a^n = \begin{cases} (a^{\frac{n}{2}})^2 & \text{when } n \text{ is even} \\ a(a^{\lfloor \frac{n}{2} \rfloor})^2 & \text{when } n \text{ is odd} \end{cases}$$

function fast_expon(a,n)

if n=0, return 1

if n=1, return a

if n is even,

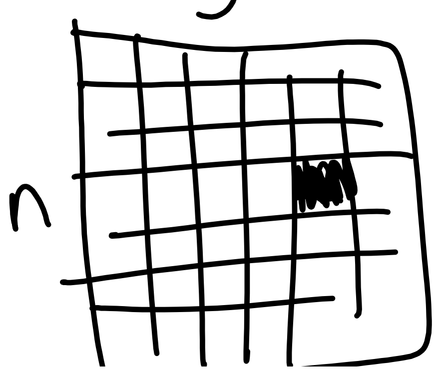
return (fast_expon(a, $\frac{n}{2}$))^2

else

return a * (fast_expon(a, $\lfloor \frac{n}{2} \rfloor$))^2

$$T(n) = c + T\left(\frac{n}{2}\right)$$
$$= c + \left(c + T\left(\frac{n}{4}\right)\right)$$
$$\vdots$$
$$= O(\log n)$$

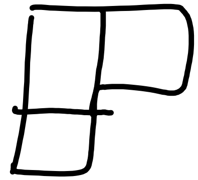
Tiling with triminoes

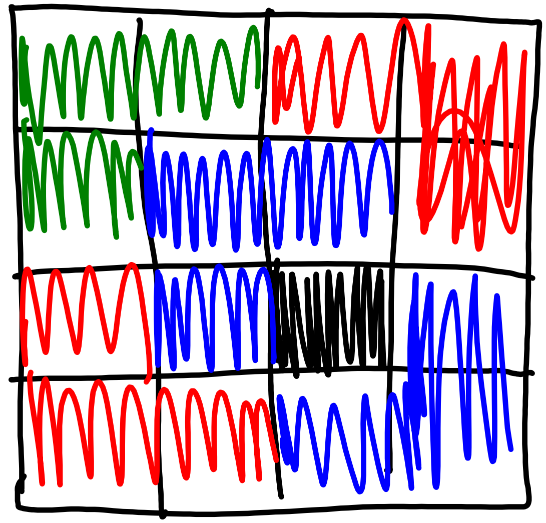


$n \times n$ checkerboard w/ 1 cell missing

$n = 2^k$, $k \in \mathbb{Z}^+$

n

 triminoes



proc