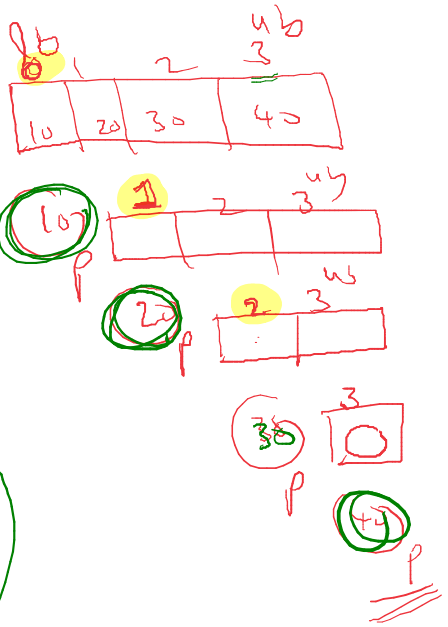
Recursion

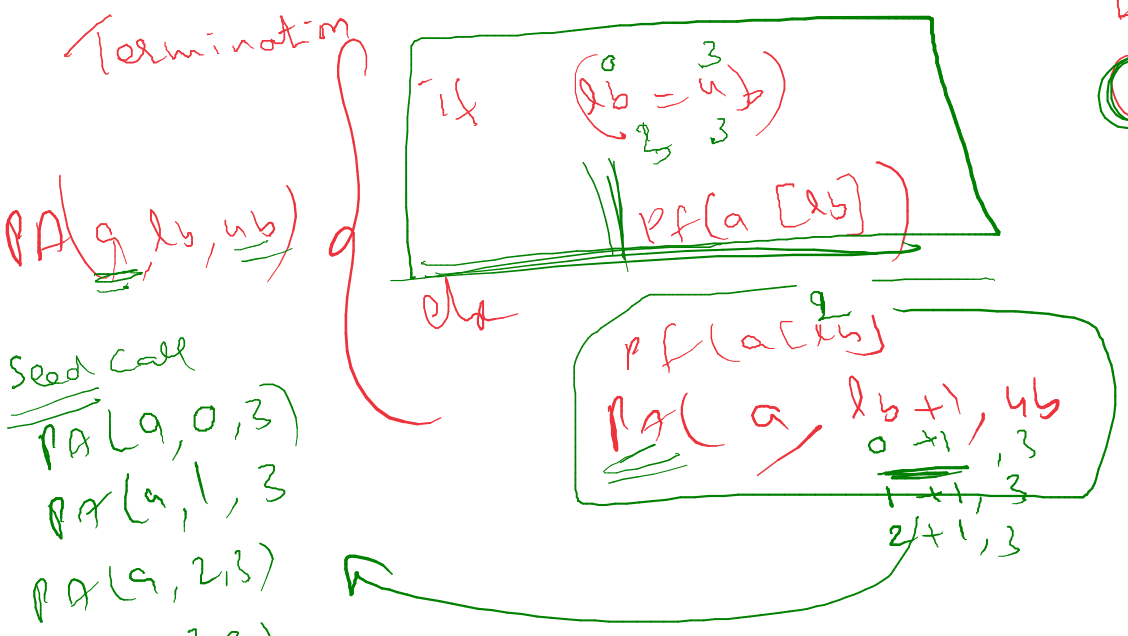
Write a function to print array elements recursively.

Properties:

1. Rec should have Termination Condition } Anchor step
2. Repeated step (Recurrence step) } Inductive step.

Tracing





Algo

PA(a, lb, ub) {

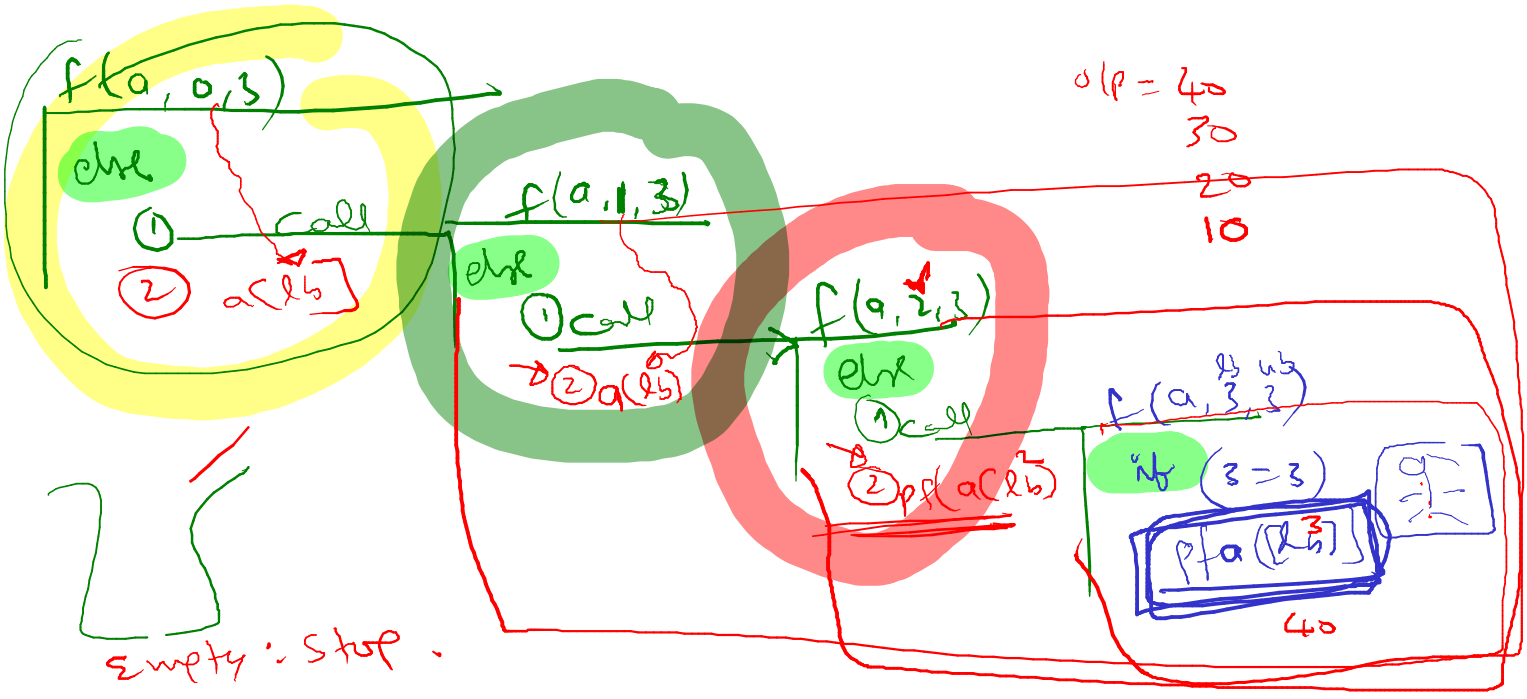
pf(a[lb])	(lb == ub) Anchor
pf(a[lb]) PA(a, lb+1, ub)	otherwise Inductive

```
void PA( int a[], int lb, int ub)
{
    if ( lb == ub)
    {
        printf("%d\n",a[lb]);
    }
    else
    {
        PA(a, lb+1,ub);
        printf("%d\n",a[lb]);
    }
}
```

f(a,0,3)
f(a,1,3)
f(a,2,3)
f(a,3,3)=40

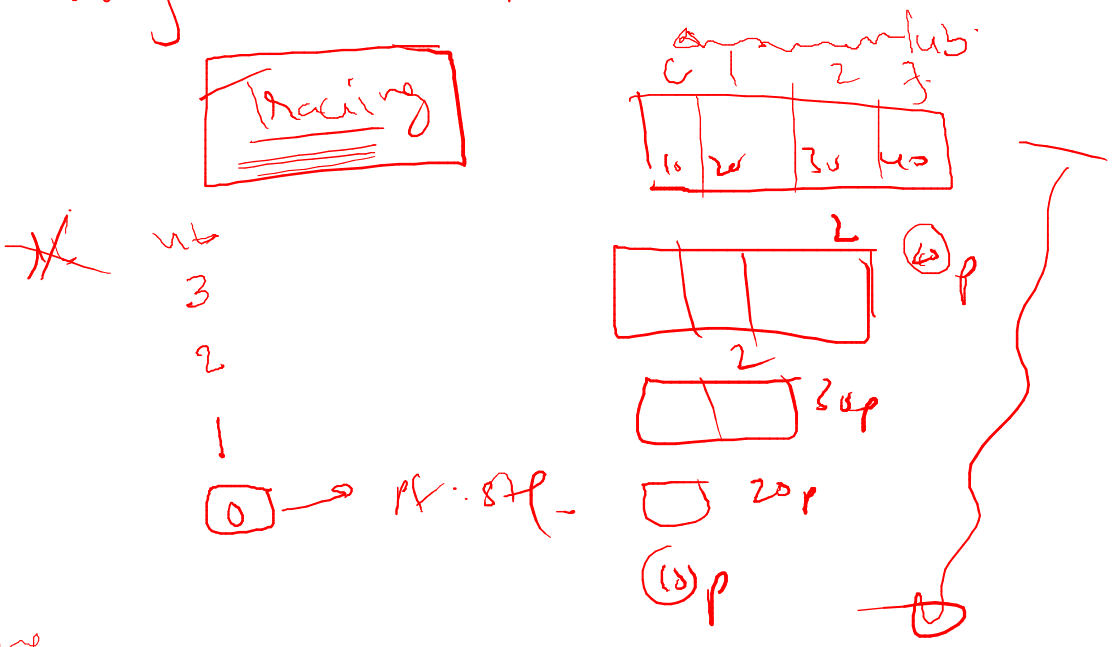
Tracing

Invocation/call

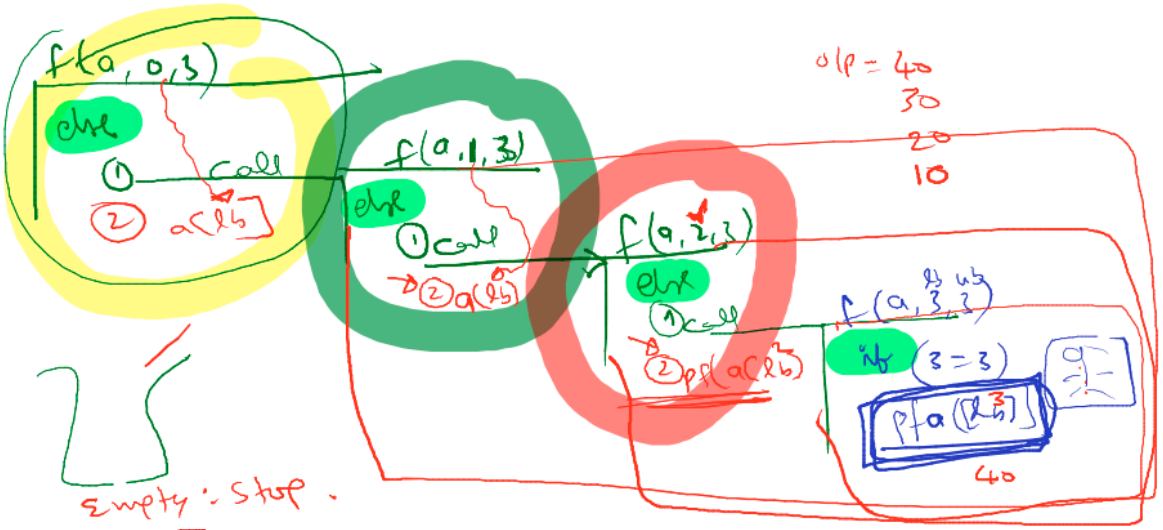


Try : Print array elements in reverse using Tail Rec.

Tracing



Tracing



Try

Q1) Search an element in an array recursively

Q2) Return min element in array recursively.

Enco

Multiplication → repeated addition

$\overset{a}{2} \times \overset{b}{3} = 6$

addition

$2 + 2 + 2 = 6$

Mul $(\overset{a}{2}, \overset{b}{3})$

2×3

$\overset{a}{2} + [\overset{a}{2} \times \overset{b}{2}]$

$\overset{a}{2} + [\overset{a}{2} \times \overset{b}{1}]$

2

Anchor

$(b = 1)$

pl(a)

$a + f(a, b-1)$

Mul → repeated +

Int DIV → repeated Subtraction

$7/2 = 3$

$6/2 = 3$

V. Imp

Nested Rec

Ackermann's function

$A(x,y)$ { $A(\ldots A(\quad))$

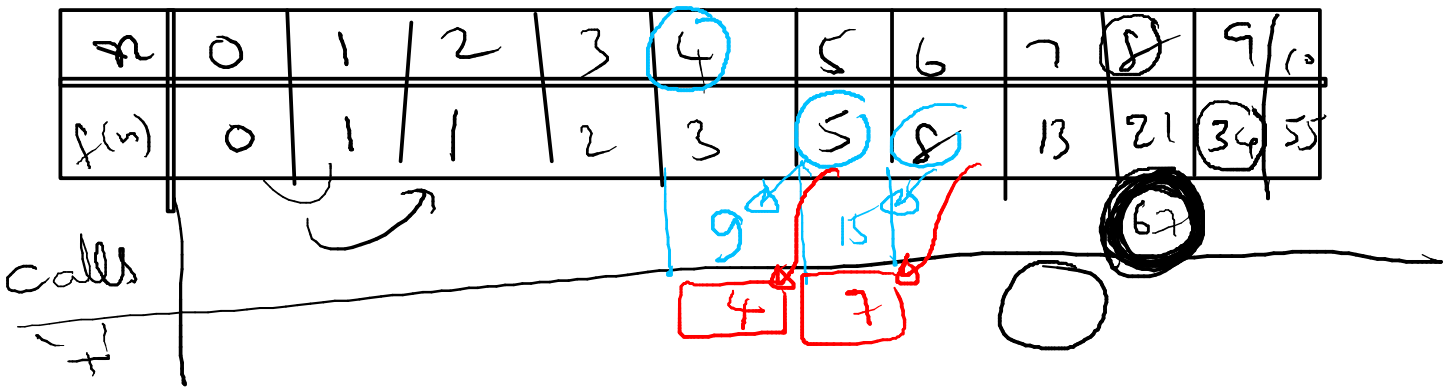
Fibonacci Series.

$$fib(n) \begin{cases} 0 & n=0 \\ 1 & n=1 \\ fib(n-1)+fib(n-2) & n>1 \end{cases}$$

Q How many calls for fib(8)

How many '+' for fib 7

n	0	1	2	3	4	5	6	7	8	9	10
f(n)	0	1	1	2	3	5	8	13	21	34	55



calls $\Rightarrow$ $2*f(N+1)-1$

call (fib5) $= 2*f(5+1)-1$

$= 2*f(6)-1$

$= 2(8)-1 = 16-1 = 15$

f(n)

div

f(n-1) + f(n-2)

n=0 => 0

n=1 => 1

mit operasonce

+ in f(n): f(n+1) - 1

+ in f(7) = f(7+1) - 1

= f(8) - 1

= 21 - 1 = 20

GCD

GCD m n (8, 21)

GCD (21, 8)

GCD (8, 5)

(5, 3)

(3, 2)

(2, 1)

(1, 0)

21)8(0
0

n 8)21(2 m
16

n 5)8(1 m
5

n 3)5(1 m
3

n 2)3(1 m
2

1)2(2
2

0)1(

GCD (21, 7)

GCD (7, 0)

n 7)21(3 m
21

0)7(

=> 7

$$GCD(m,n) = \begin{cases} m & n=0 \quad \text{(Anchor)} \\ GCD(n,\ m\%n) & \text{others} \end{cases}$$

$$GCD(a,b) = \begin{cases} a & b=0 \\ GCD(b,\ a\%b) & \end{cases}$$

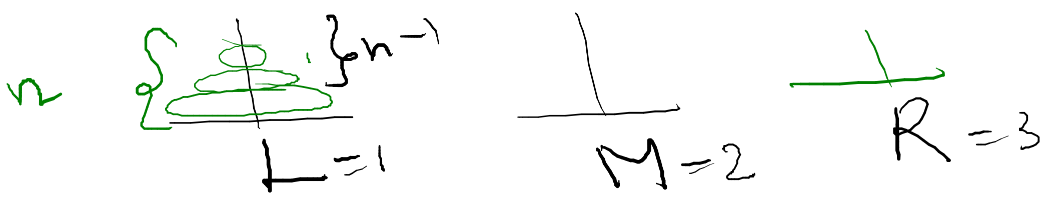
GAME

Rule: only one disc

Recursively

Towers of Hanoi

20 Q

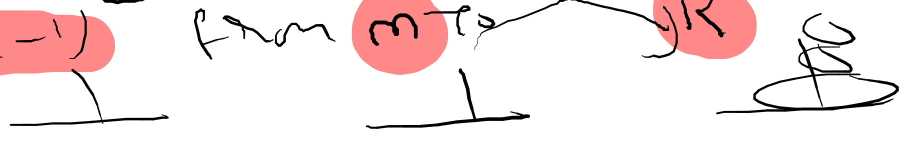


Algo

① lift (n-1) discs from L to M

② move 1 from L to R

③ lift (n-1) from M to R



$$fac(N) = \begin{cases} 1 & N = 0 \\ N \times fac(N-1) & \text{otherwise} \end{cases}$$

$$f(n) = n \times f(n-1)$$

$$f(4) = 4 \times f(3)$$

$$3 \times f(2)$$

$$2 \times f(1)$$

$$1 \times f(0)$$

$$1$$

$$4 \times 3 \times 2 \times 1$$

Algo

ToH(n, L, M, R)

If (n != 0)

{

① ToH(n-1, L, R, M)

② Move(1/d L:R)

③ ToH(n-1, M, L, R)

}

Seed
T(3, 1, 2, 3)

T(n, L, M, R)

1

T(n-1, L R M)

L M

T(n-1, M L R)

TRACING

Call by value

T(3, 1, 2, 3)

T(2 L R M)
1 3 2
L M R

L M
1 3

T(2, M L R)
2 1 3
L M R

T(1 L R M)
1 2 3
L M R

L M
1 2

T(1, M L R)
3 1 2
L M R

T(1, L R M)
2 3 1
L M R

L M
2 3

T(1, M L R)
1 2 3
L M R

T(0 ...) L M (0 ...)
1 3

T(0 ...) L M (0 ...)
3 2

T(0 ...) L M (0 ...)
2 1

T(0 ...) L M (0 ...)
1 3

Moves

1

2

3

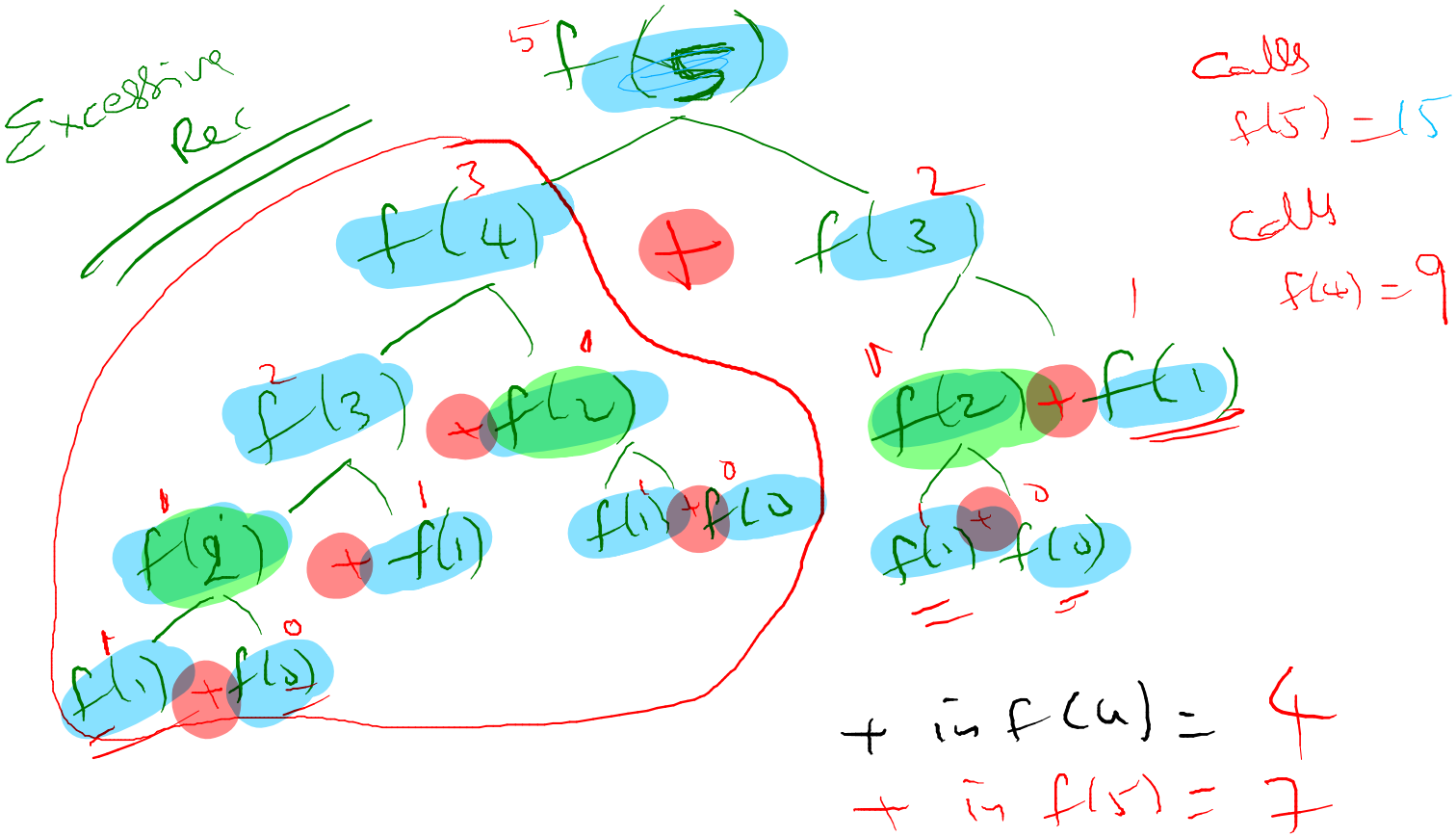
4

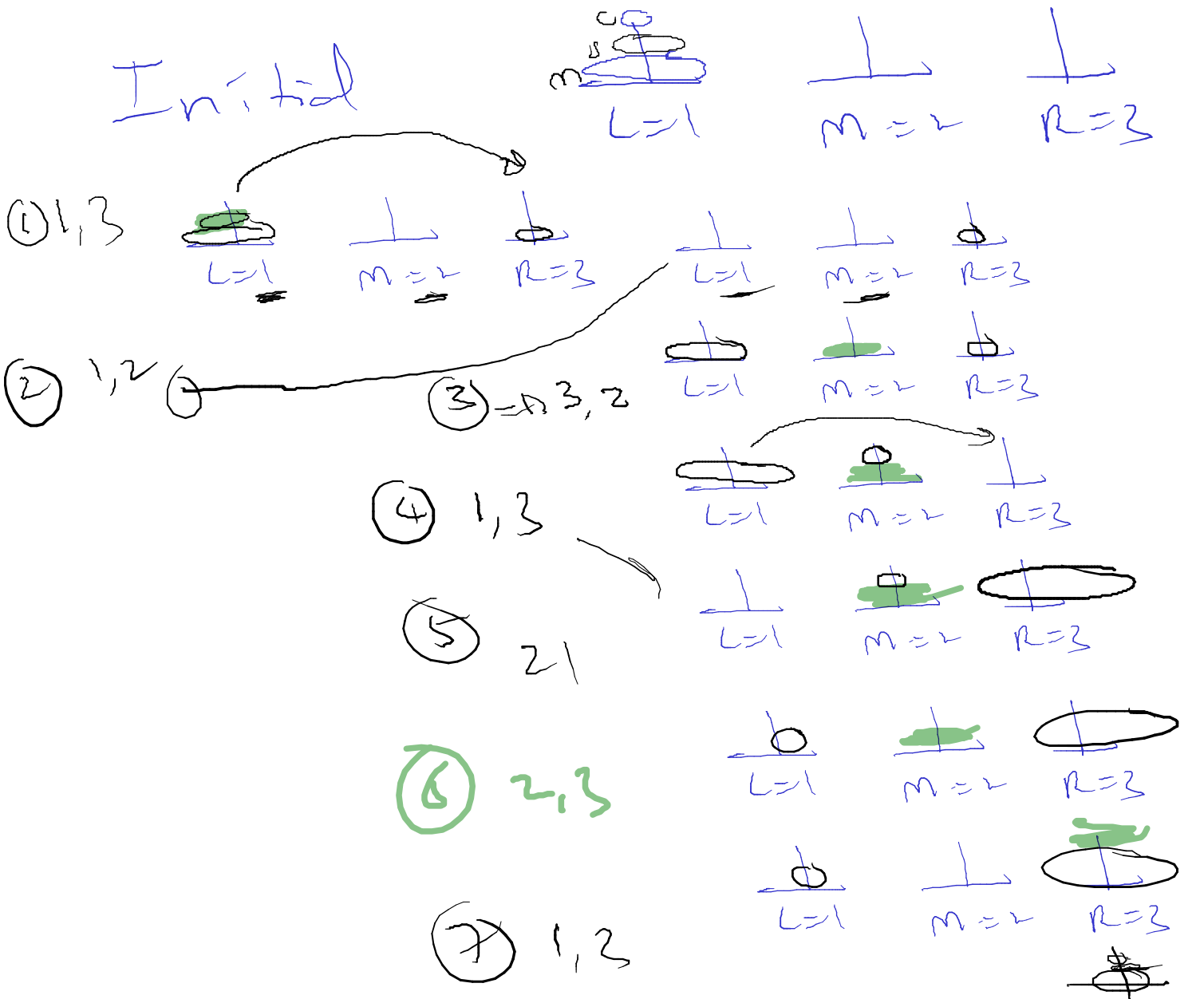
5

6

7

L=1 M=2 R=3


Excessive Rec
f(5)
f(4)
f(3)
f(3)
f(2)
f(1)
f(2)
f(1)
f(1)
f(0)
f(1)
f(0)
f(1)
f(0)
Calls
f(5) = 15
Calls
f(4) = 9
+ in f(4) = 4
+ in f(5) = 7


Initial
L=1
M=2
R=3
1) 1,3
2) 1,2
3) 3,2
4) 1,3
5) 21
6) 2,3
7) 1,3