~~P2~~ (Pms 2 u)

```
tetramultiple(n)

if (n==0 V n==1 V n==2)
    return 0

if (n==3)
    return 1

~~else~~
    return (tetramultiple(n-1) + tetramultiple(n-2)
+ tetramultiple(n-3) + tetramultiple(n-4))
```

43 2 1 6 7 8 5

L : 5

F : 54

F(3,2,1): 12354

L(6,7,8) 12354678

0 0 0 1 1 2 4 8 15 29

Part 2a)

```
tetraRecursive (n, a, b, c, d)

    if (n==0)
        return a
    if (n==1)
        return b
    if (n==2)
        return c
    if (n==3)
        return a+b+c+d

    return (n-1, a+b+c+d, a, b, c)

tetra(n){
    return tetraRecursive(n, 0, 0, 0, 1)
}
```

3

B(6) F(1,2,3,4

~~remove~~

Q.enque(D.remove

Q.enque(D.remove

~~D.enque(~~

D.addFirst(Q.deque()

D.offerFirst(Q.de

D.offerFirst(

n = 7

call (6, 1, 0, 0, 0) 2

call (5, 1, 1, 0, 0) 1

call (4, 2, 1, 1, 0) 1

call (3, 4, 2, 1, 1)

Part 2

c)

Linear

$t(5,0,0,0,1)$

$\downarrow$

$t(4,1,0,0,0)$ ← return 2

$\downarrow$

$t(3,1,1,0,0)$ ← return 2

No branching. Parameters a, b, c, d are incremented, and the are returned upon reaching base case. Since parameters are incremented w/ each call, repeated calculations are not a problem. algorithm essentially performs ~ n-2 recursive calls. So, algorithm is $\Theta(n)$

Multiple

$t(5)$

$t(4)$ $t(3)$ $t(2)$ $t(1)$

$t(3)$ $t(2)$ $t(1)$ $t(0)$

Multiple branches due to 4 recursive calls each time (it has as base case). From tree structure on left, it can be observed thus each recursive call may result in 4^n function calls at most. So, this algorithm is $\Theta(4^n)$

Possible Bottleneck: doing subproblem again. e.g $T(n)$ will call $T(n-1)$, $T(n-2)$, $T(n-3)$, $T(n-4)$

~~and $T(n-1)$ will call $T(n-2)$, $T(n-$~~

and $T(n-1)$ will call $T(n-2)$, $T(n-3)$, $T(n-4)$, $T($

As can be seen, same subproblem are repeated with each recursive call.

d) LinearRecursive function uses tail recursion, as tail recursion is defined as a single recursive call at end of function, which matches structure of fibonacciLinearRecursive().

n vs. Execution Time for Linear and Multiple Recursive Tetranacci Calculator

n	t_1	t_2
0	600	1100
5	300	700
10	400	5700
15	800	124400
20	800	316900
25	1400	5901300
30	3300	173755200

$t_1 \sim mn + b$

REGRESSION PARAMETERS

$m = 76.42857$ $b = -60.71429$

STATISTICS

$R^2 = 0.6302$

$r = 0.7938$

RESIDUALS

e_1 plot

$t_2 \sim a*b^{(n)}$

Log Mode

REGRESSION PARAMETERS

$a = 0.268942$ $b = 1.96642$

STATISTICS

$R^2 = 1$

RESIDUALS

e_2 plot