

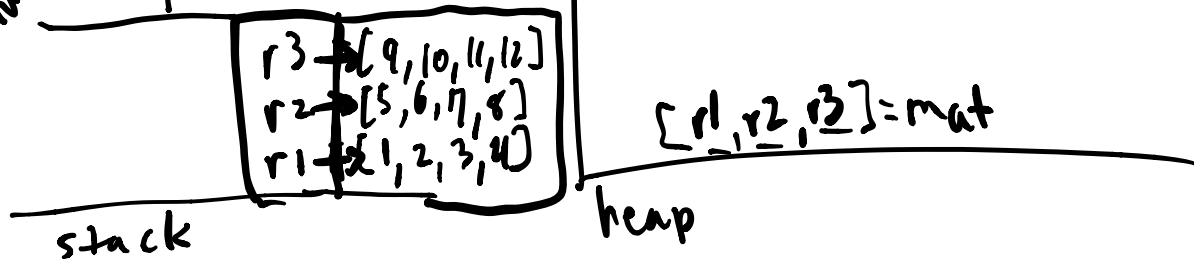
Recursive call untill $n == 0$.

ret val = $r[0] = 1$
func mat $[r_1, r_2, r_3]$ $n=0$ $m=4$

func mat $[r_1, r_2, r_3]$ $n=1$ $m=4$

func mat $[r_1, r_2, r_3]$ $n=2$ $m=4$

func mat $[r_1, r_2, r_3]$ $n=3$ $m=4$



Return from $h == 1$
 (changes are marked in red)

func2
 ret val = 10
 + emp = $0 + 1 + 2 + 3 + 4 = 10$
 in = r1 | [1, 2, 3, 4] x = 4
 ret val = 10
 mat[0][0] = r1[0] ← 1
 func1 mat | [r1, r2, r3] n = 1 m = 4

func1 mat | [r1, r2, 3] n = 2 m = 4

func1 mat | [r1, r2, r3] n = 3 m = 4

r3	[9, 10, 11, 12]
r2	[5, 6, 7, 8]
r1	[2, 3, 4]

[r1, r2, r3] = mat

stack

heap

Return from $n == 2$

func2 $retval = 31$
temp = $0 + 10 + 6 + 7 + 8 = 31$
in = r2 [10, 6, 7, 8] $x = 4$

$retval = 31$
 $mat[1][0] = r2[0] \leftarrow 10$

func1 $mat[r1, r2, 3]$ $n = 2$ $m = 4$

func1 $mat[r1, r2, r3]$ $n = 3$ $m = 4$

$r3$	\rightarrow	[9, 10, 11, 12]
$r2$	\rightarrow	[10, 6, 7, 8]
$r1$	\rightarrow	[1, 2, 3, 4]

stack

heap

$[r1, r2, r3] = mat$

Return from $n == 3$

func² ret val = 64
temp = 0 + 31 + 10 + 11 + 12 = 64
in = r3 \rightarrow [31, 10, 11, 12]

ret val = 64
mat[2][0] = r3[0] \leftarrow 31
func mat[r1, r2, r3] n=3 m=4

r3	\rightarrow [31, 10, 11, 12]
r2	\rightarrow [16, 6, 7, 8]
r1	\rightarrow [1, 2, 3, 4]

stack

[r1, r2, r3] = mat

heap

Rest of the main (the for loop)

