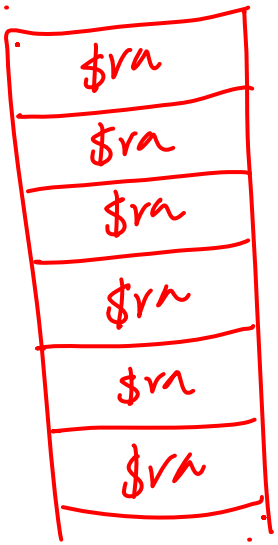
$a0 [1]

```
bunnyEars:
        beq $a0, $0, baseCase
        addi $sp, $sp, -4
        sw $ra, 0($sp)
        addi $a0, $a0, -1
        jal bunnyEars
        addi $v0, $v0, 2
        lw $ra, 0($sp)
        addi $sp, $sp, 4
        jr $ra
baseCase:
        add $v0, $0, $0
        jr $ra
```

saves $ra to stack

```
int bunnyEars(int bunnies) {
    if(bunnies == 0)
        return 0;
    else
        return 2 + bunnyEars(bunnies - 1);
}
```

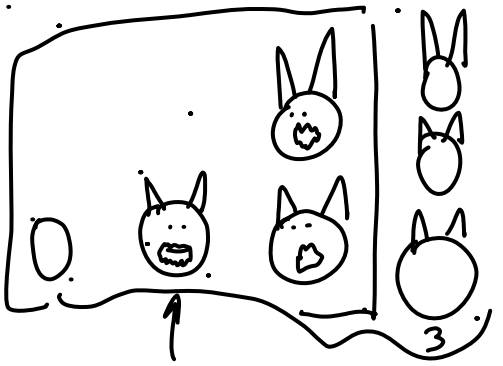


Key points

- use $v0 for return value
- save $ra for non-leaf
- save registers you might need

Recursive example:

Compute the sum of 0 to n



Alien Bunny 3

```
int Sum(int n) {
    if(n == 0)
        return 0;
    else
        return (n + Sum(n-1));
}
```

($a0)

- Save to stack: $ra, $a0

```
Sum:
    beq $a0, $0, baseCase
    addi $sp, $sp, -8
    sw $ra, 0($sp)
    sw $a0, 4($sp)
    addi $a0, $a0, -1
    jal Sum
    lw $a0, 4($sp)
    lw $ra, 0($sp)
    add $v0, $a0, $v0
    addi $sp, $sp, 8
    jr $ra

baseCase:
    add $v0, $0, $0
    jr $ra
```

Sum(n-1) → $v0

$a0 = n

MIPS addressing modes

MIPS addressing modes

MIPS addressing modes