

Day 11

## 2/21/25 Recursion Problems

Print  
N times  
 $O(N)$

```
void f(i, n) {
    if (i > n)
        return;
    print —;
    f(i+1, n); // will continue to call f till untrue
               // making a "recursion tree"
}

int n = # of times;
int i = 1;
```

Print  
Linearly  
from  
1 to N

same as above but (int i = 1;) outside of void. Inside  
print i.

N →

```
use (n == 0) {
    n--;
}
```

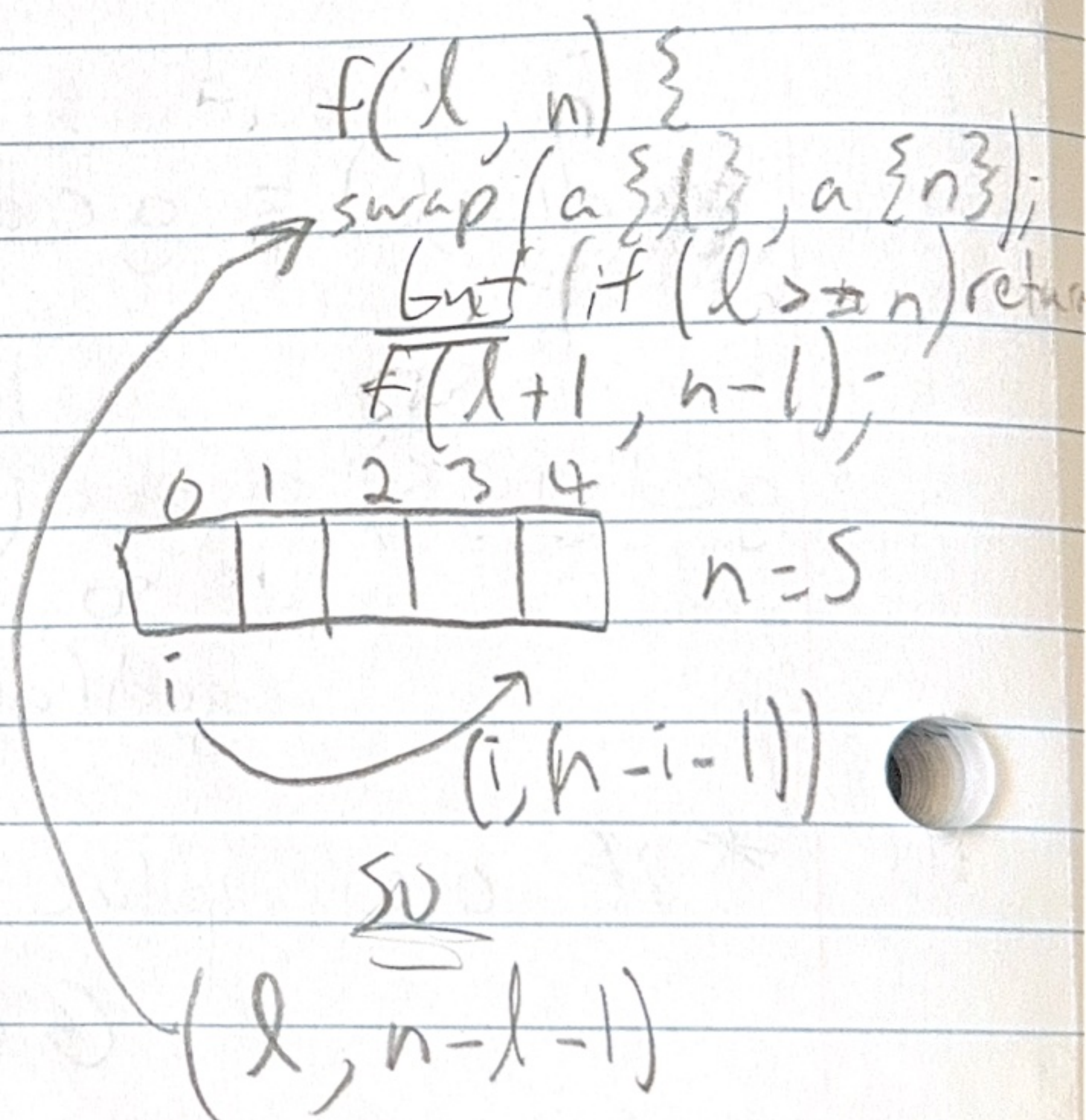
Parametrized  
f Method

```
f(i, sum) {
    if (i < 1) print sum & return;
    f(i-1, sum+i);
}
```

Functional  
Method

```
f(n)
if (n == 0) return 0;
return n + f(n-1);
}
```

### Reverse On an Array



In Problems  
Set i to be  
static if  
its not  
a parameter



2/22/25

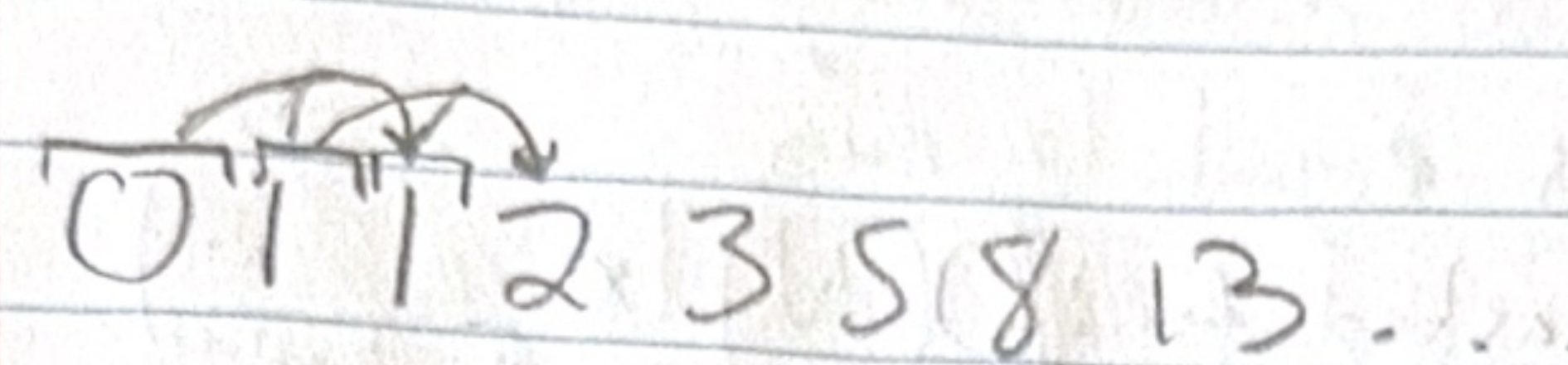
Checking M A D A M \*left and right must be same

For  
Palindrome

```
f(i) {
    if (i >= n/2) return true;
    if (s[i] != s[n-i-1])
        return false;
```

```
return f(i+1);
```

Fibonacci



\* adds current and last number

Num

```
f(n) {
    if (n <= 1)
        return n;
```

```
last = f(n-1); // first last
slast = f(n-2); // second last
or return last + slast;
```

```
return (f(n-1) + f(n-2));
```

Recursion  
Tree

before ending previous recursions, the next will never occur

