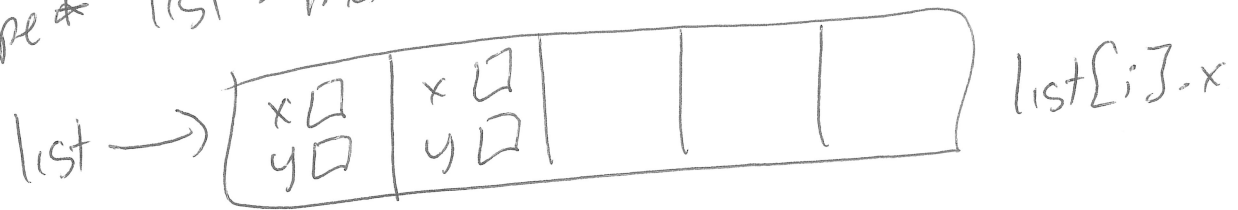
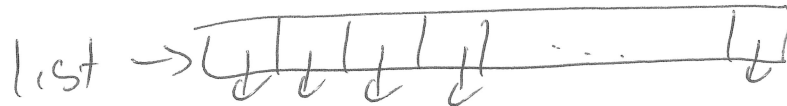


1/21/22

$\text{type}^* \text{list} = \text{malloc}(\text{sizeof}(\text{type}) * n);$

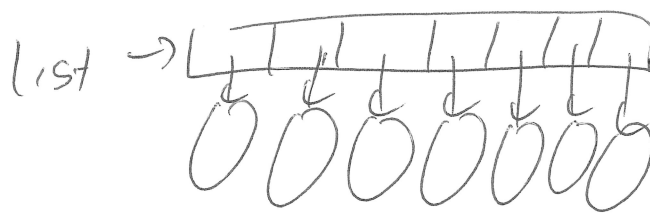


$\text{type}^* \text{list} = \text{malloc}(\text{sizeof}(\text{type}) * n);$



for (int i = 0; i < n; i++)

list[i] = malloc(sizeof(type));



list[i] $\rightarrow$ x

Recursion

A function defined by another instance of itself.

function that calls itself sometimes.

```
int funcA(int n) {  
    return funcA(n-1);
```

BAD

```
}  
NEVER END!
```

```
int funcA(int n) {  
    if (n <= 0) return 0;  
    return 1 + funcA(n-1);  
}
```

Math Recursive func

$$f(0) = 0$$

$$f(n) = 1 + f(n-1)$$

$$f(n) = 1 + f(n-1)$$

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

$$= 1 + 1 + 1 + f(n-3)$$

$$= \underbrace{1 + 1 + 1 + 1 + \dots}_{n \text{ 1's}} + f(0)$$

$$= n$$

$$g(n) = 0$$

$$g(n) = n + g(n-1)$$

$$g(n) = n + g(n-1)$$

$$= n + (n-1) + g(n-2)$$

$$\begin{matrix} & \bullet & \\ & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{matrix} \quad = n + (n-1) + (n-2) + g(n-3)$$

$$= \underbrace{n + (n-1) + \dots + 1}_{n(n+1)/2} + g(0)$$

$$= \frac{n(n+1)}{2}$$

Fib Inefficient

