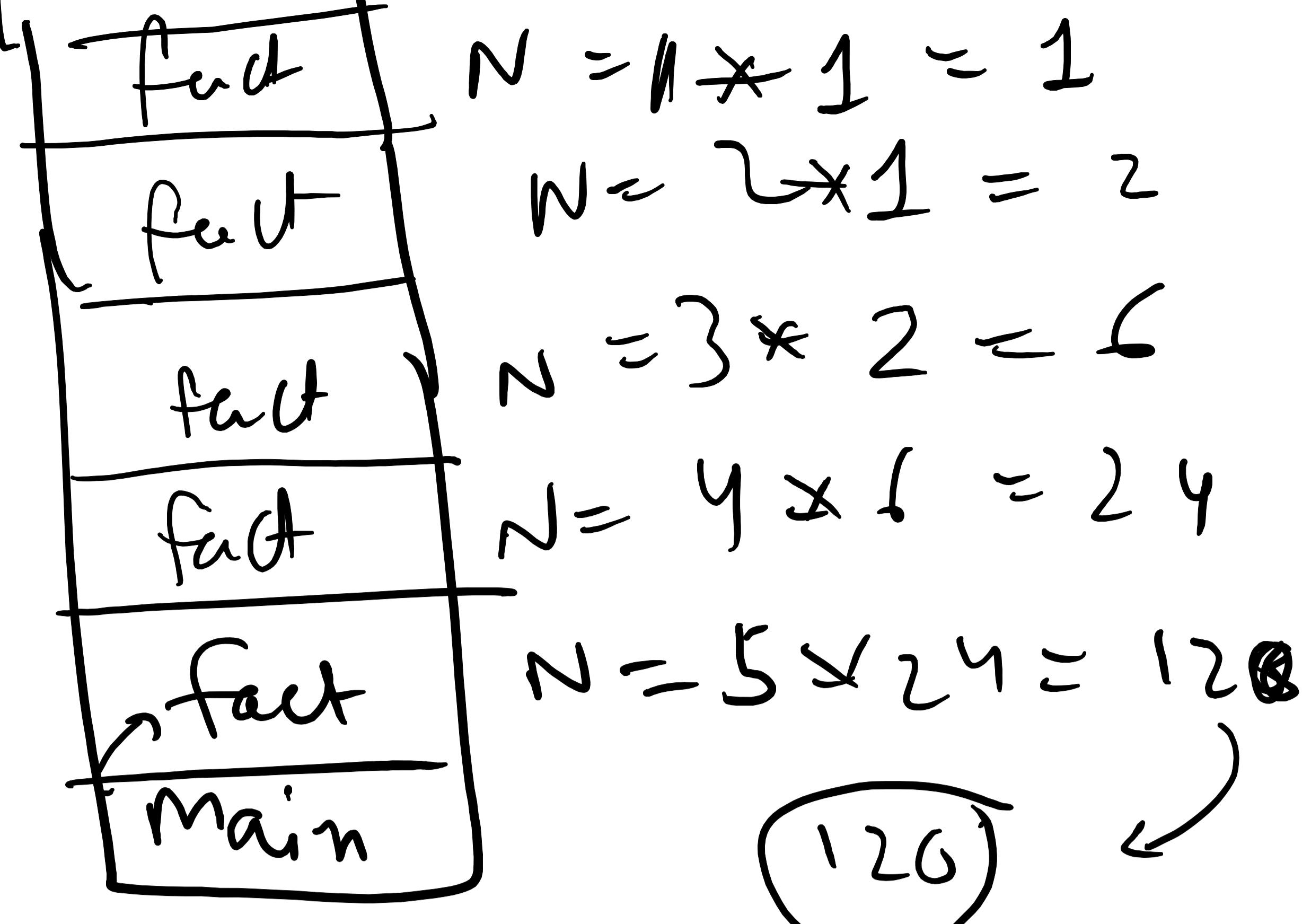


```
int fact (int n) {
    if (n == 0)
    if (n == 0)
        return 1;
```

main()

fact(5)

)



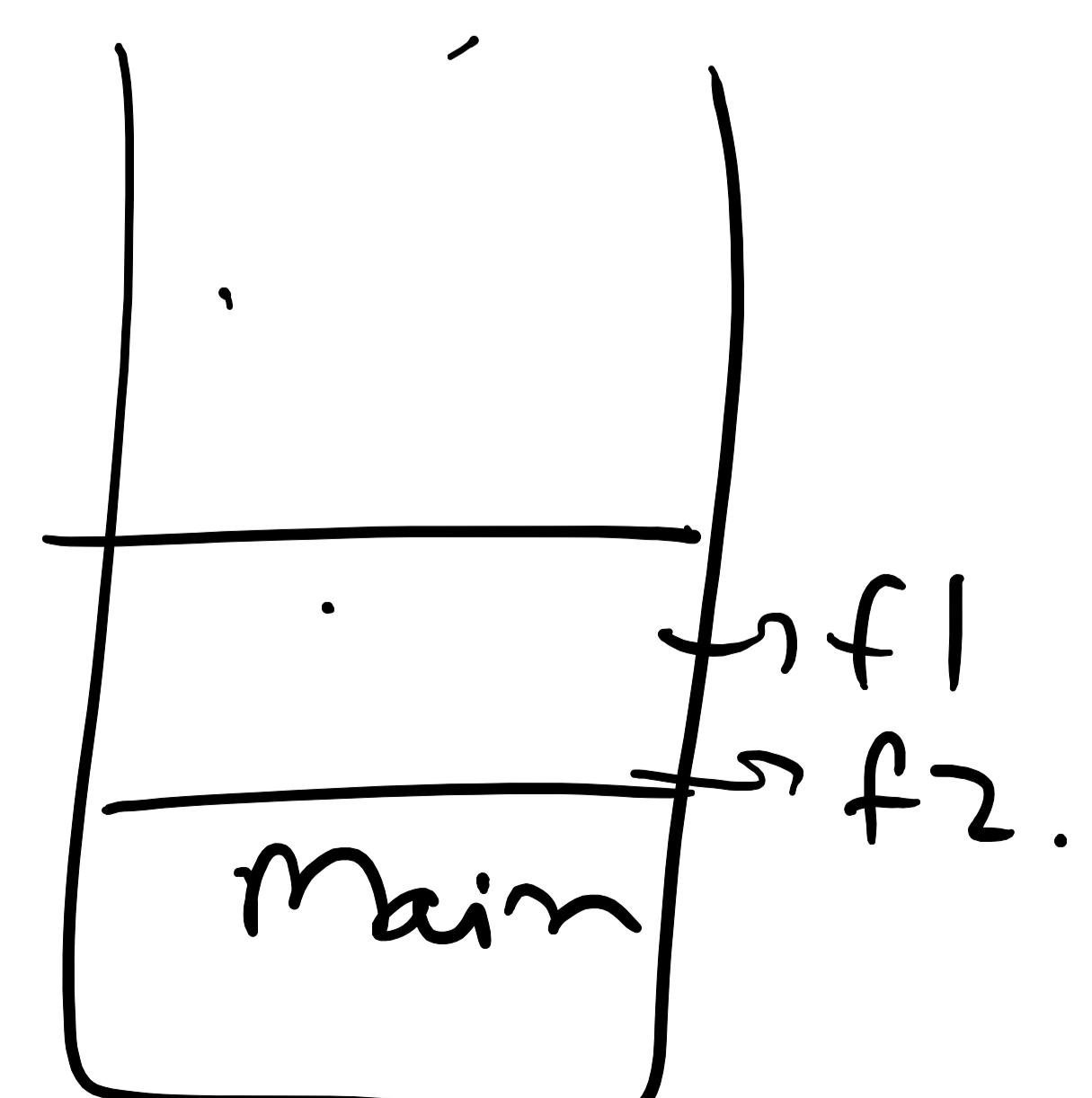
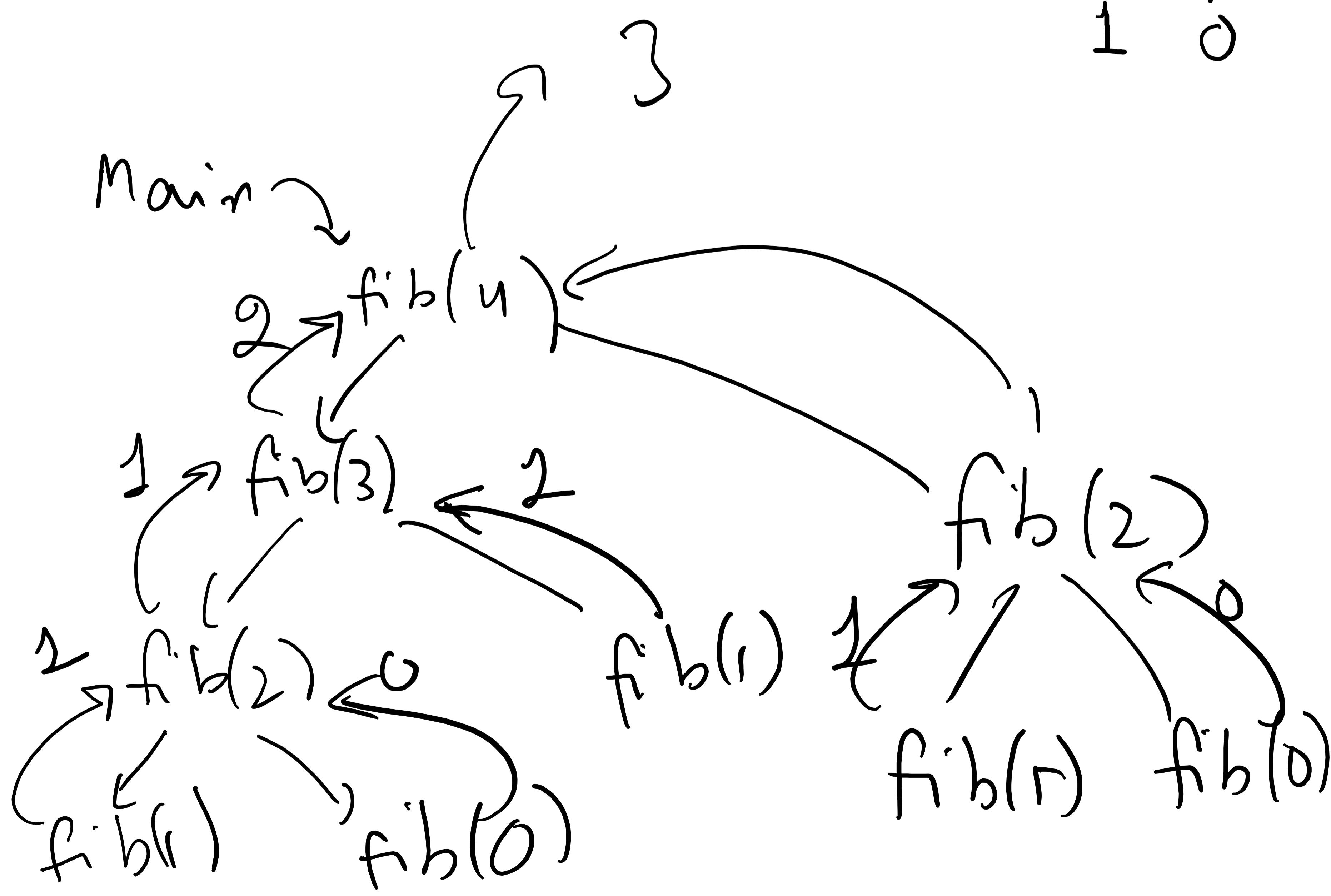
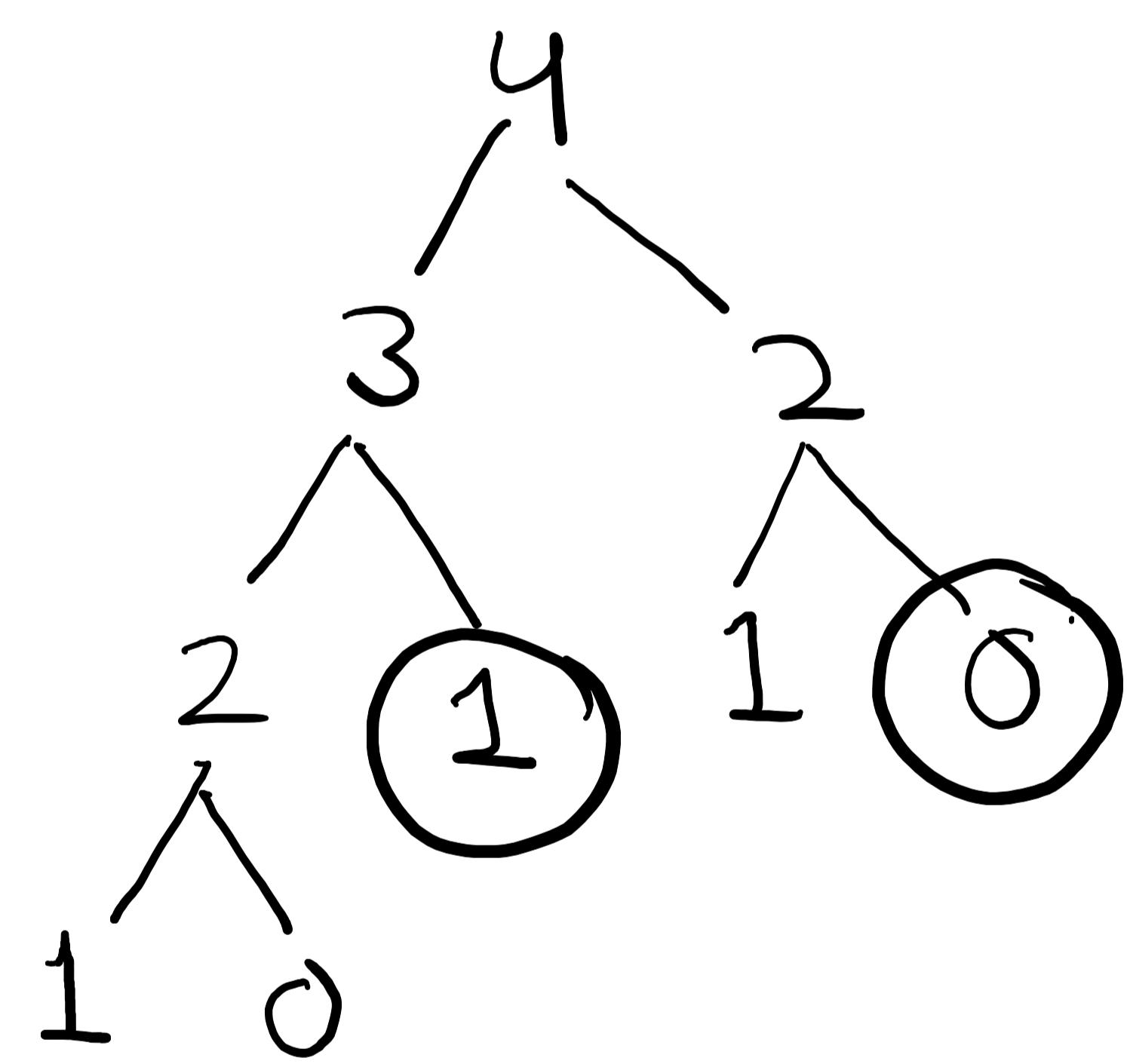
$$\begin{matrix} 5 \\ P \\ \downarrow \\ 5 \times 4! \\ 0 \end{matrix}$$

Base case

Fibonacci

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

0 1 1 2 3 5 8
... : : : : : :

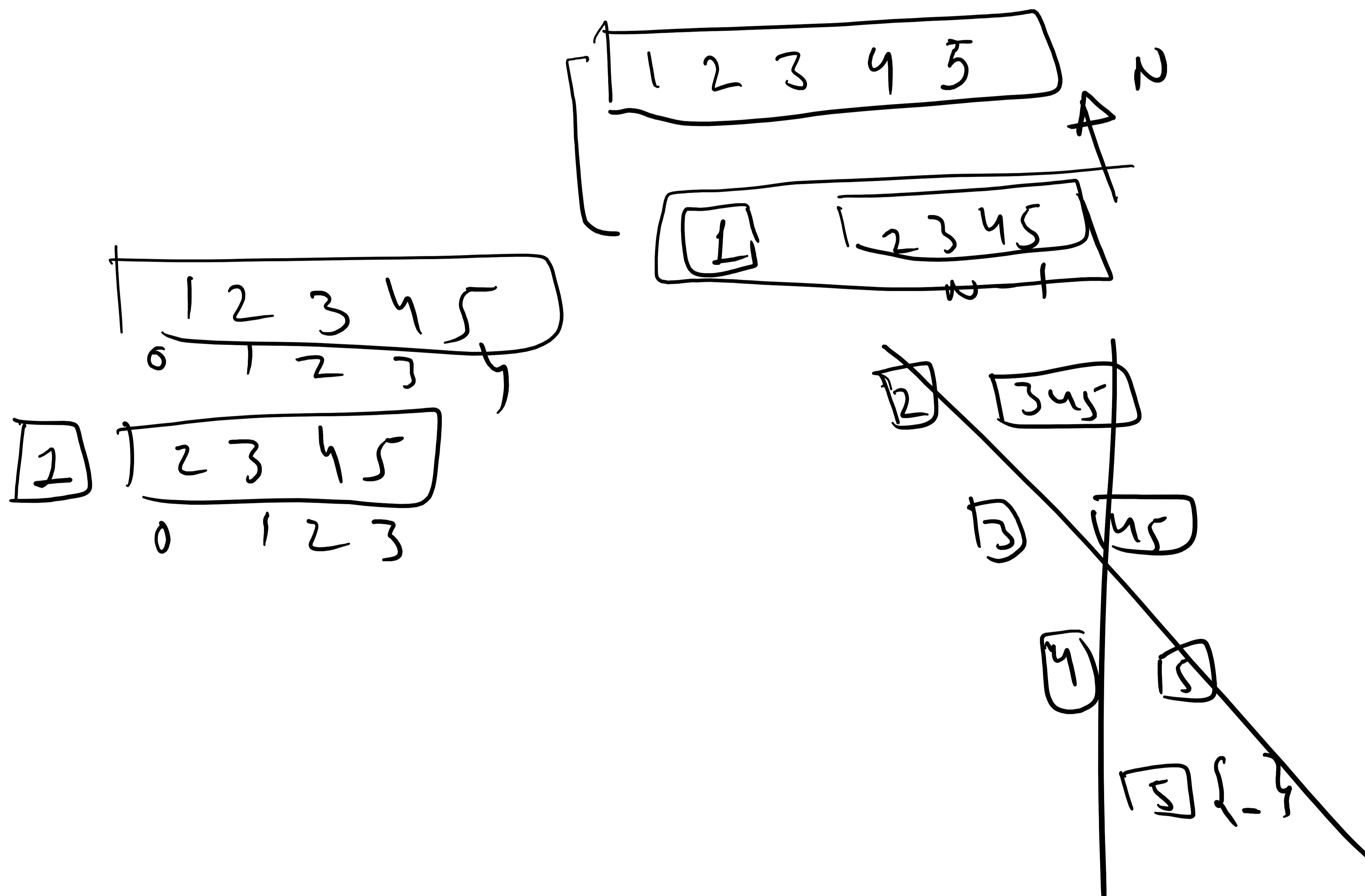


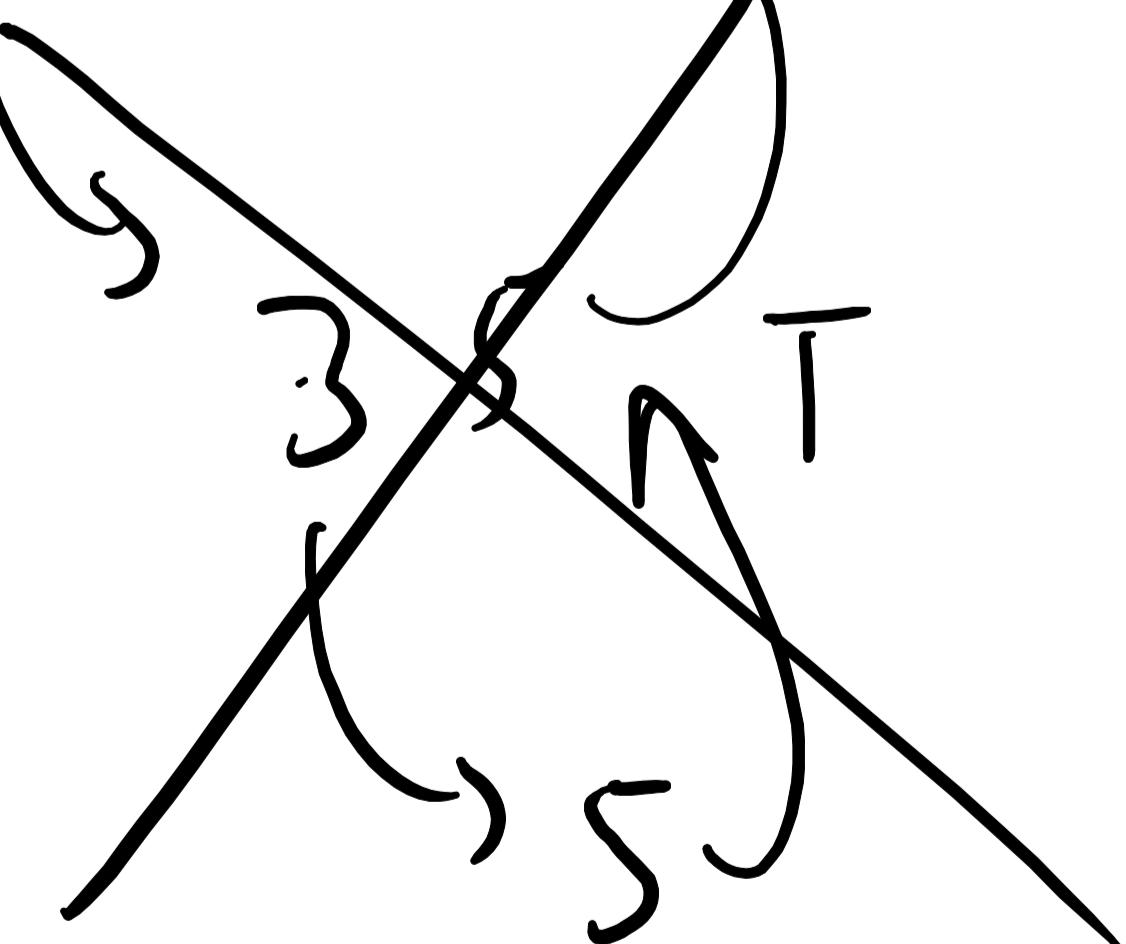
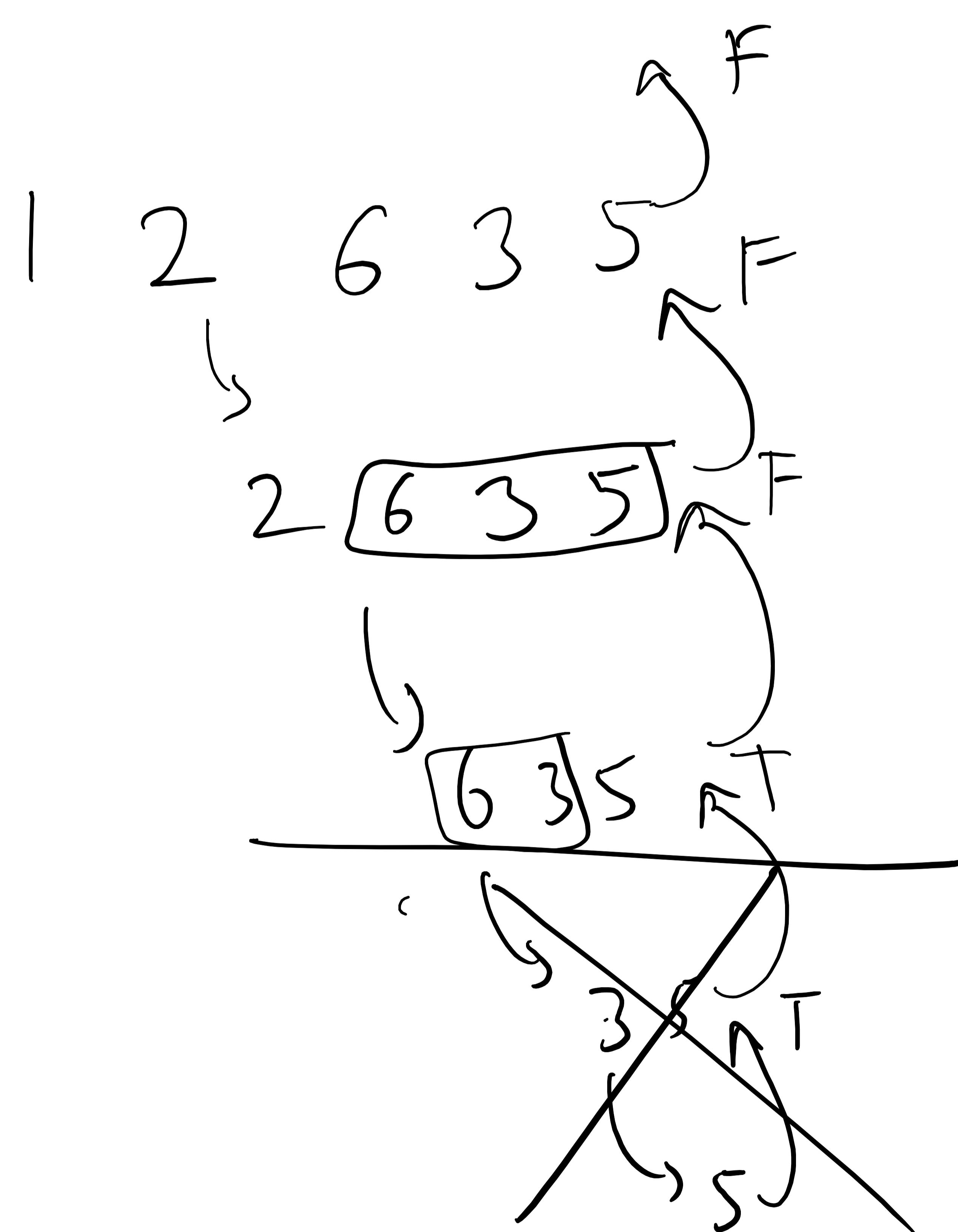
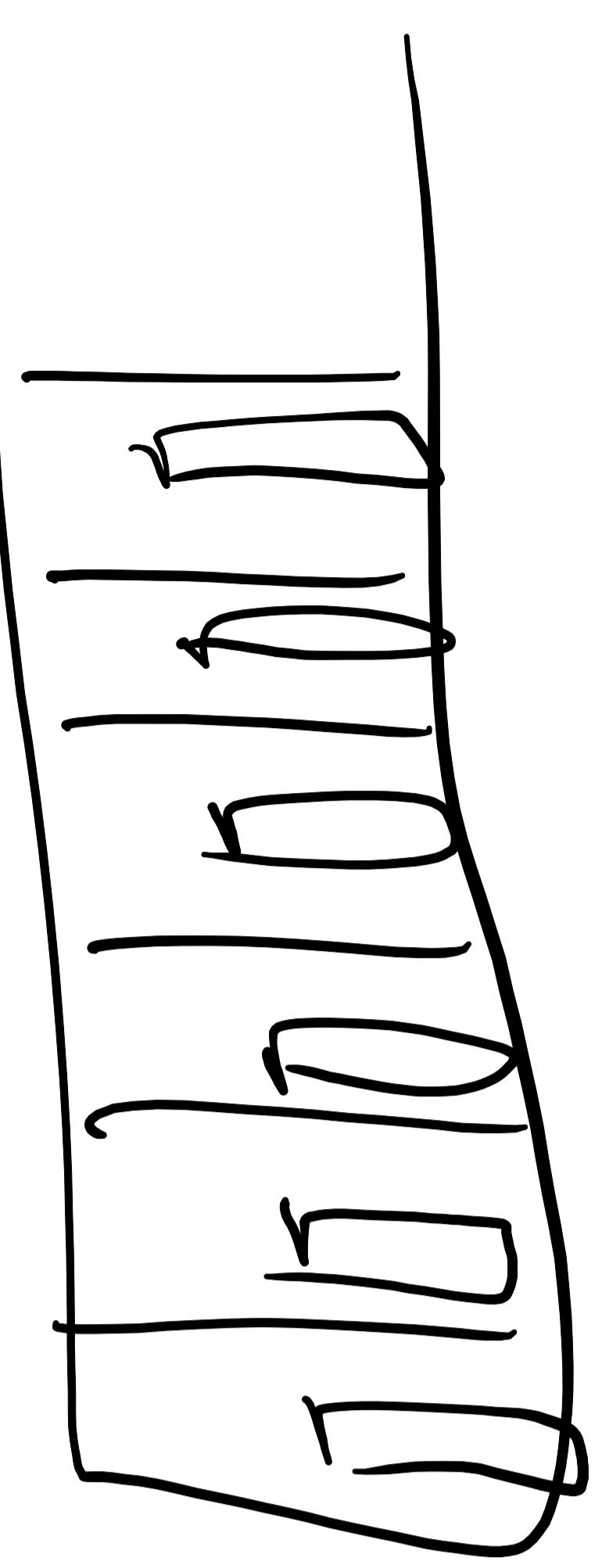
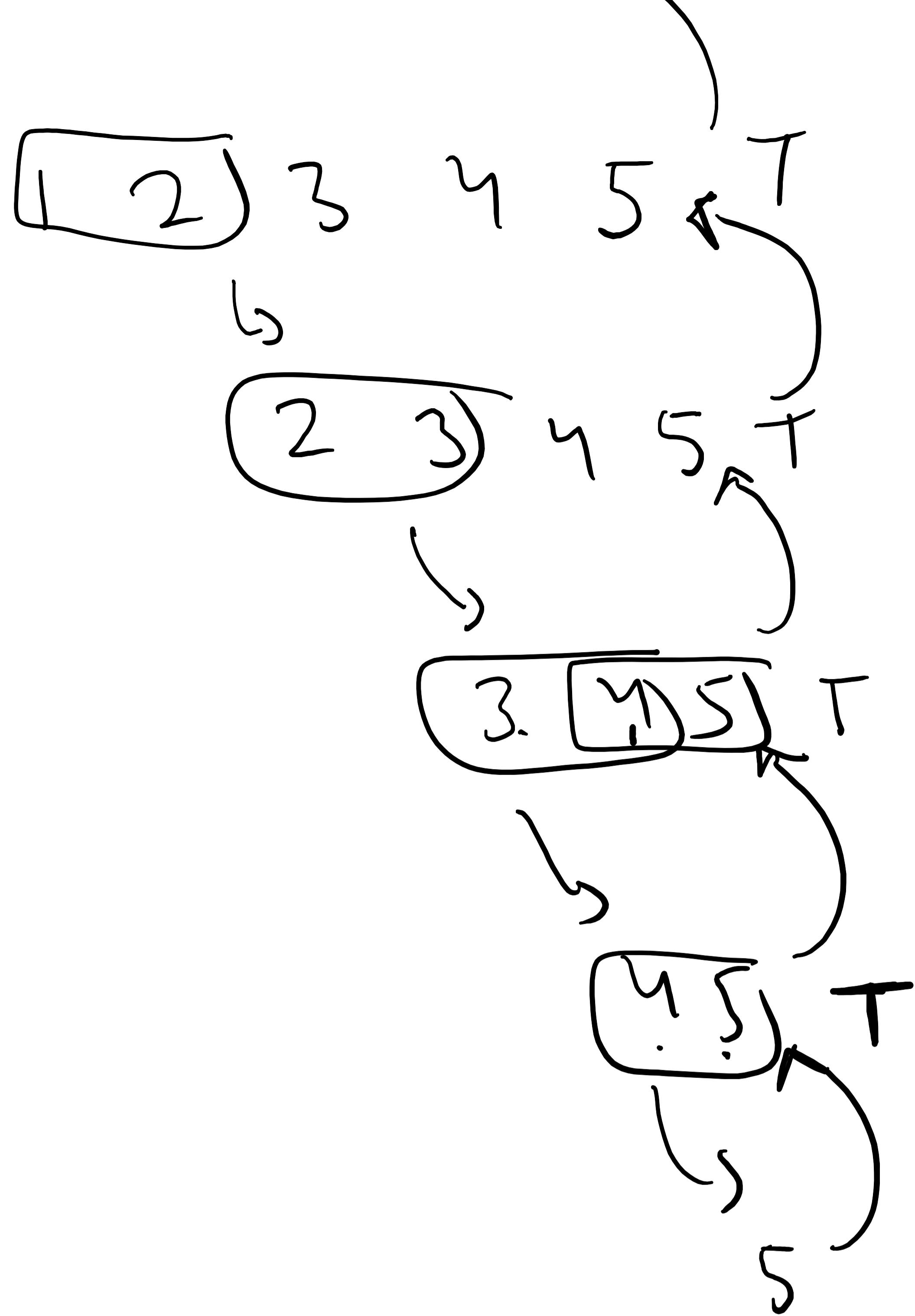
~~✓~~ Prove $k=0$ True.

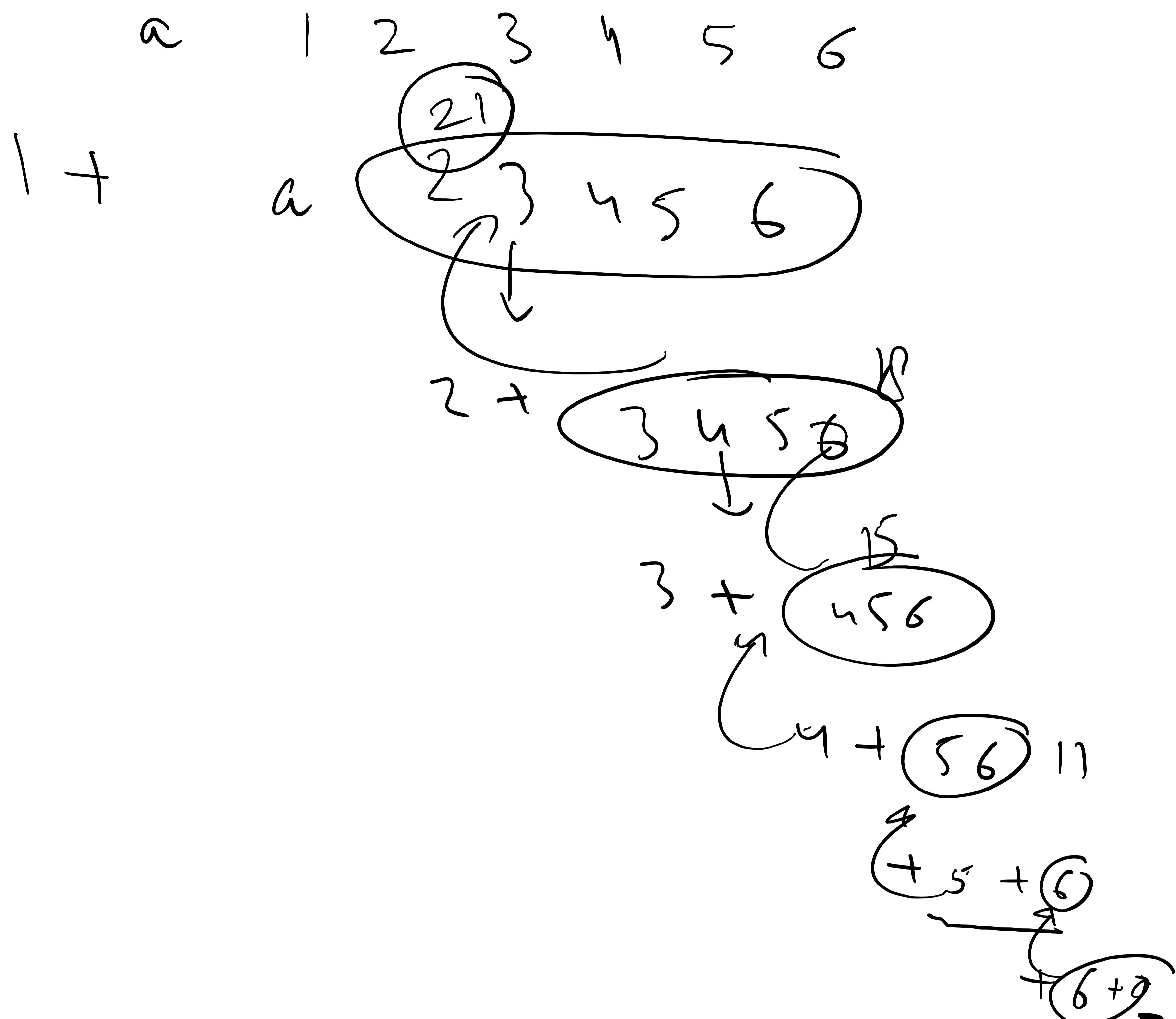
→ ② Assume $f(k-1)$ True

~~✓~~ Prove $f(k)$ is True

③ Check if a given array is SORTED (or Not)







$a[0] +$

$$n = 3$$

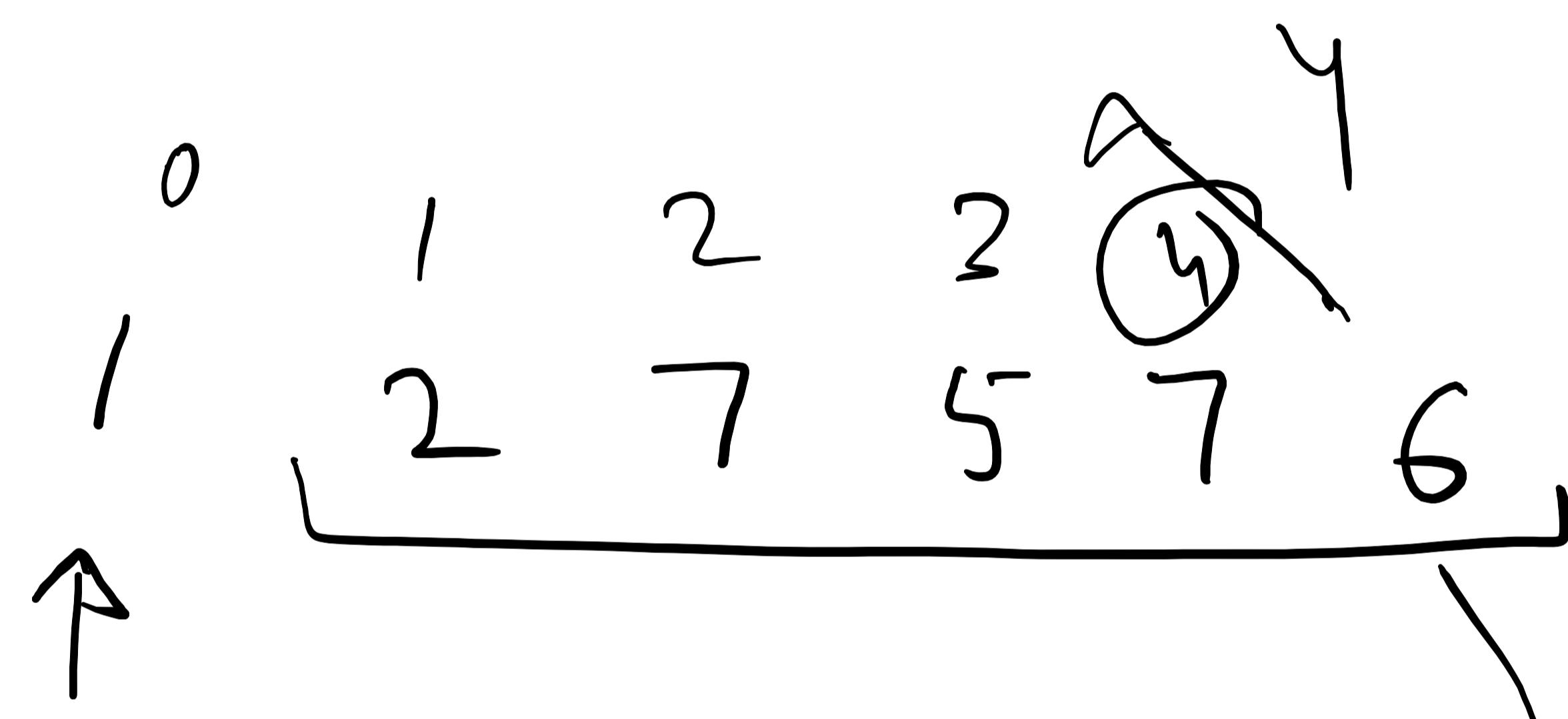
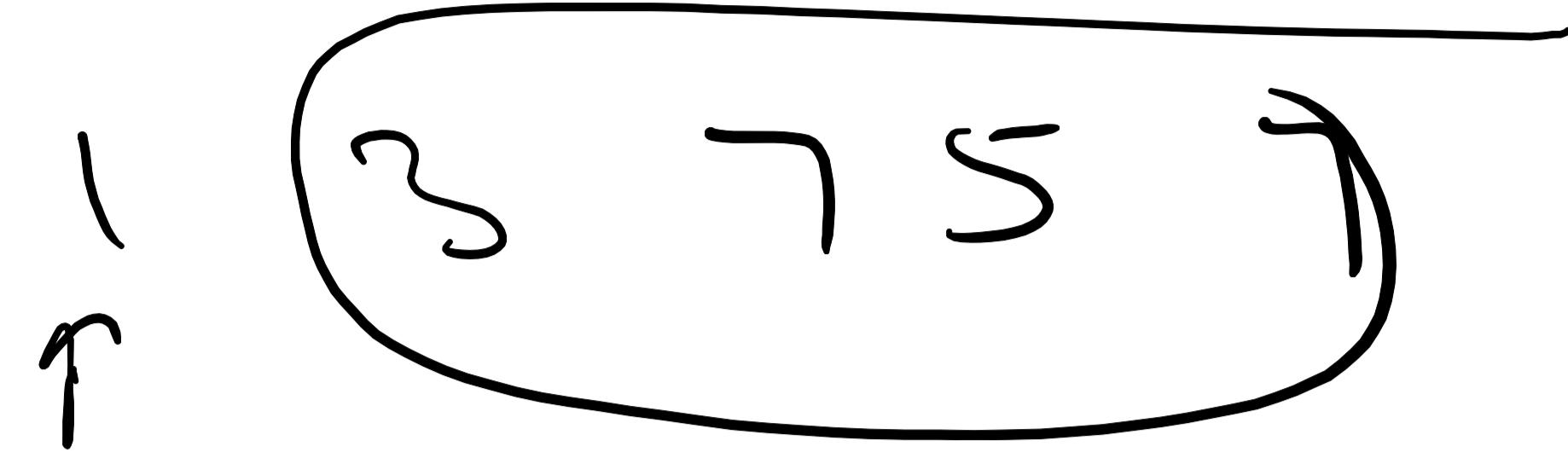
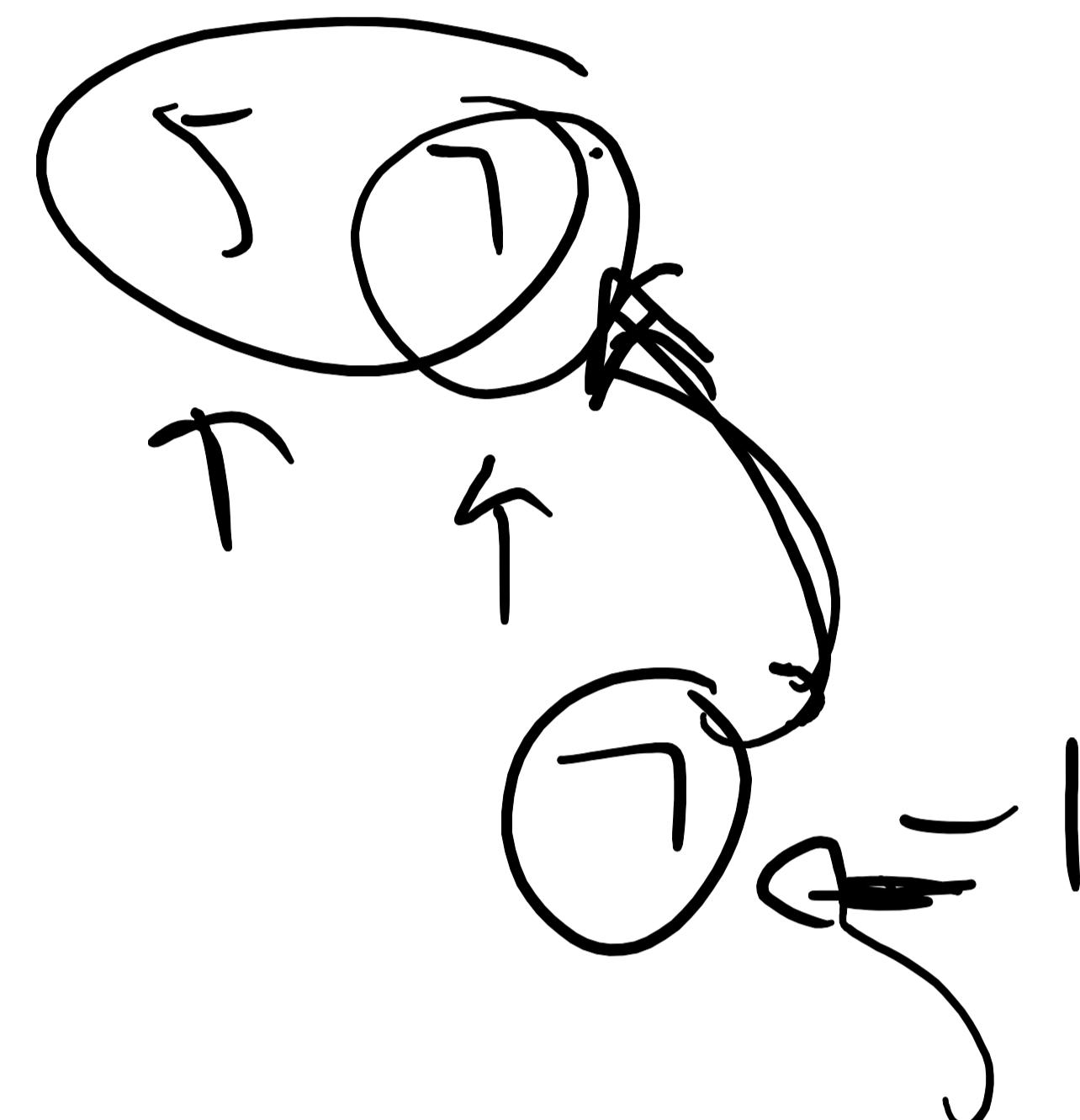
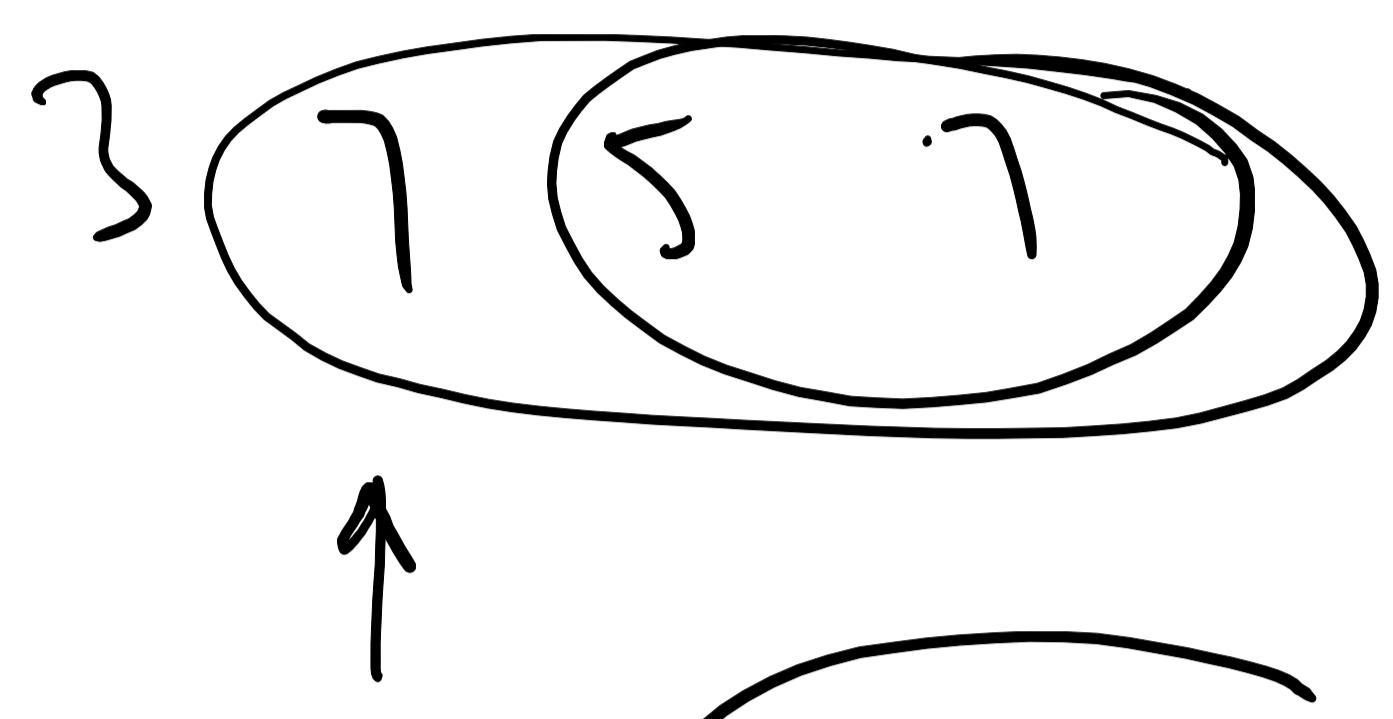
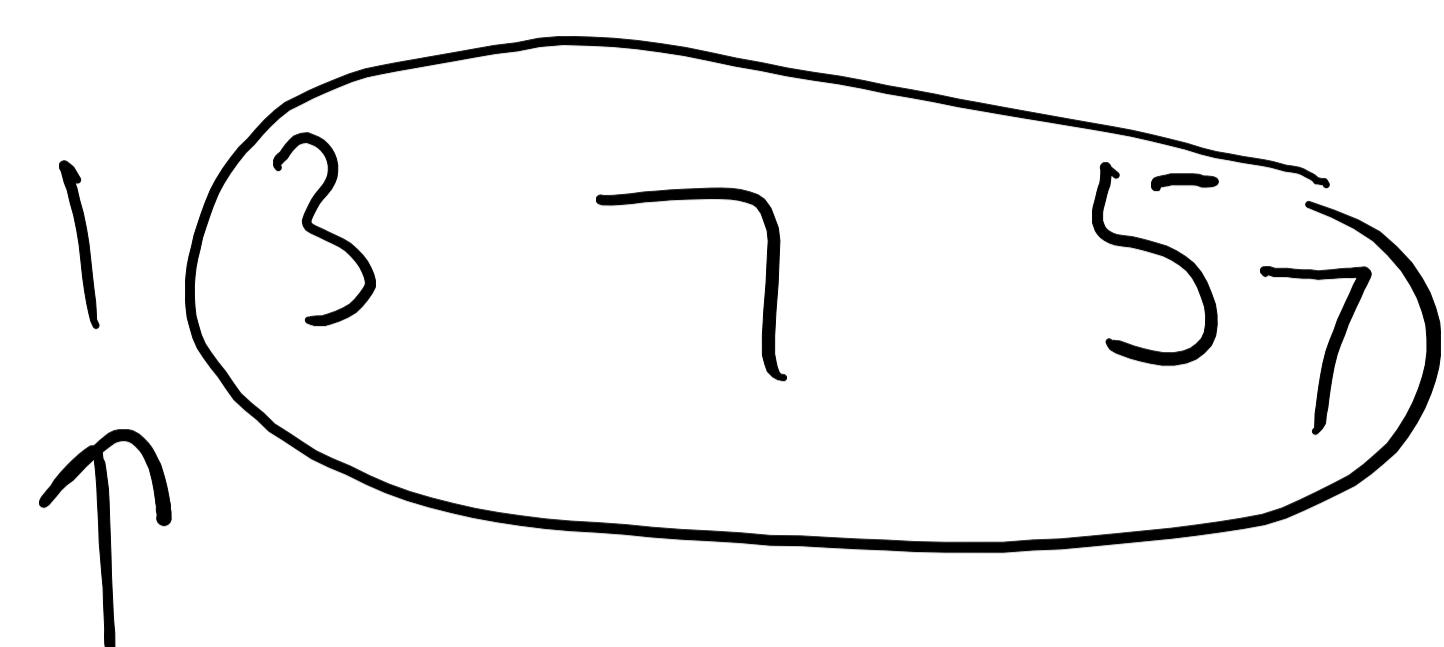
$$(1 2 3) 4$$

$$(1, 2, 3) + 4$$

$$(1 2) + 3 + 4$$

$$(1) + 2 + 3 + 4$$

$$0 + 1 + 2 + 3 + 4$$



2 7 5 7 6 4

7 5 7 6 4

5 7 6 4

7 6

6
 $i = n - 1$

$$x^y = x \cdot (\underline{x^{y-1}})$$

int power(x, y)

if (y == 0)

return 1;

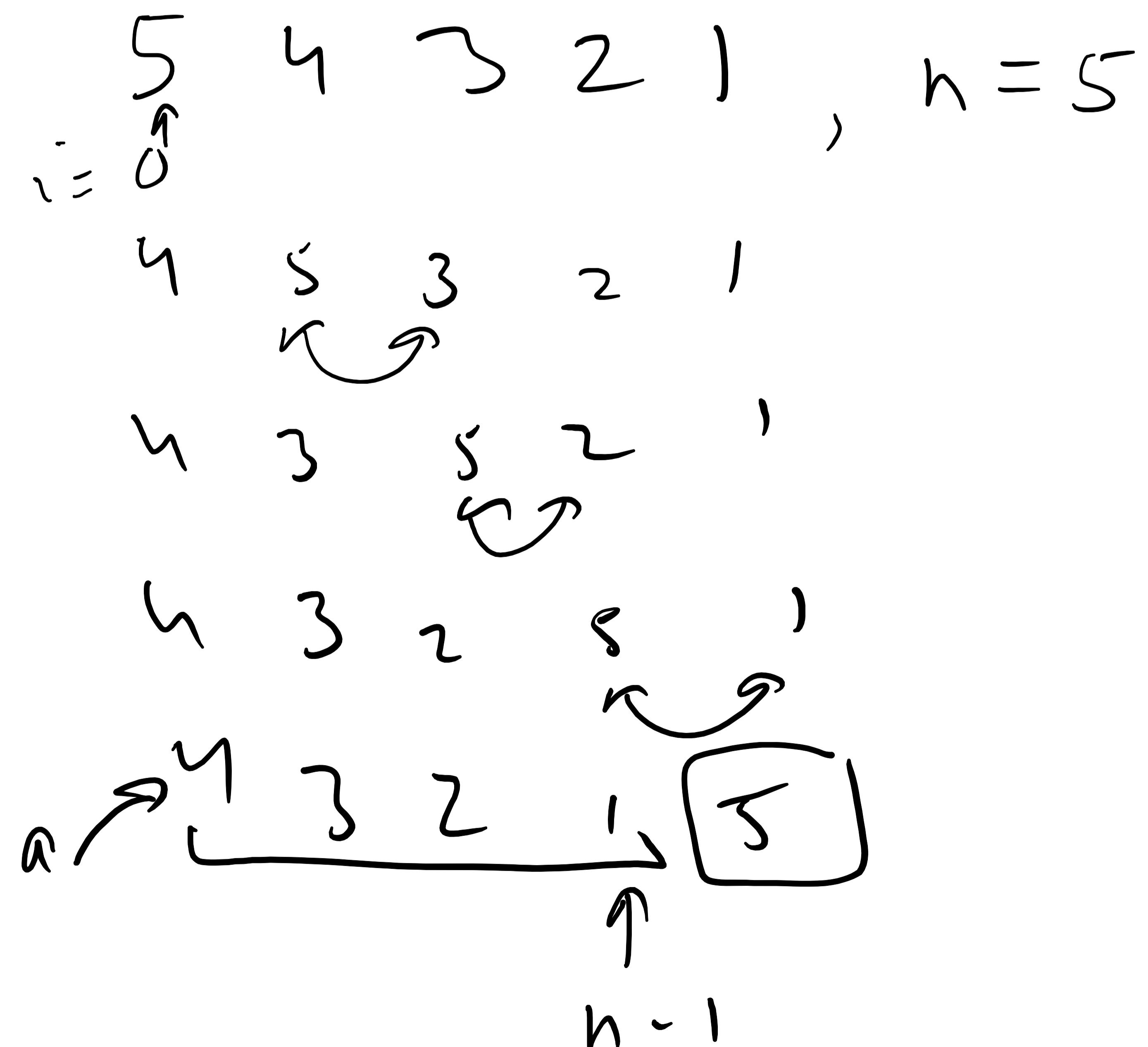
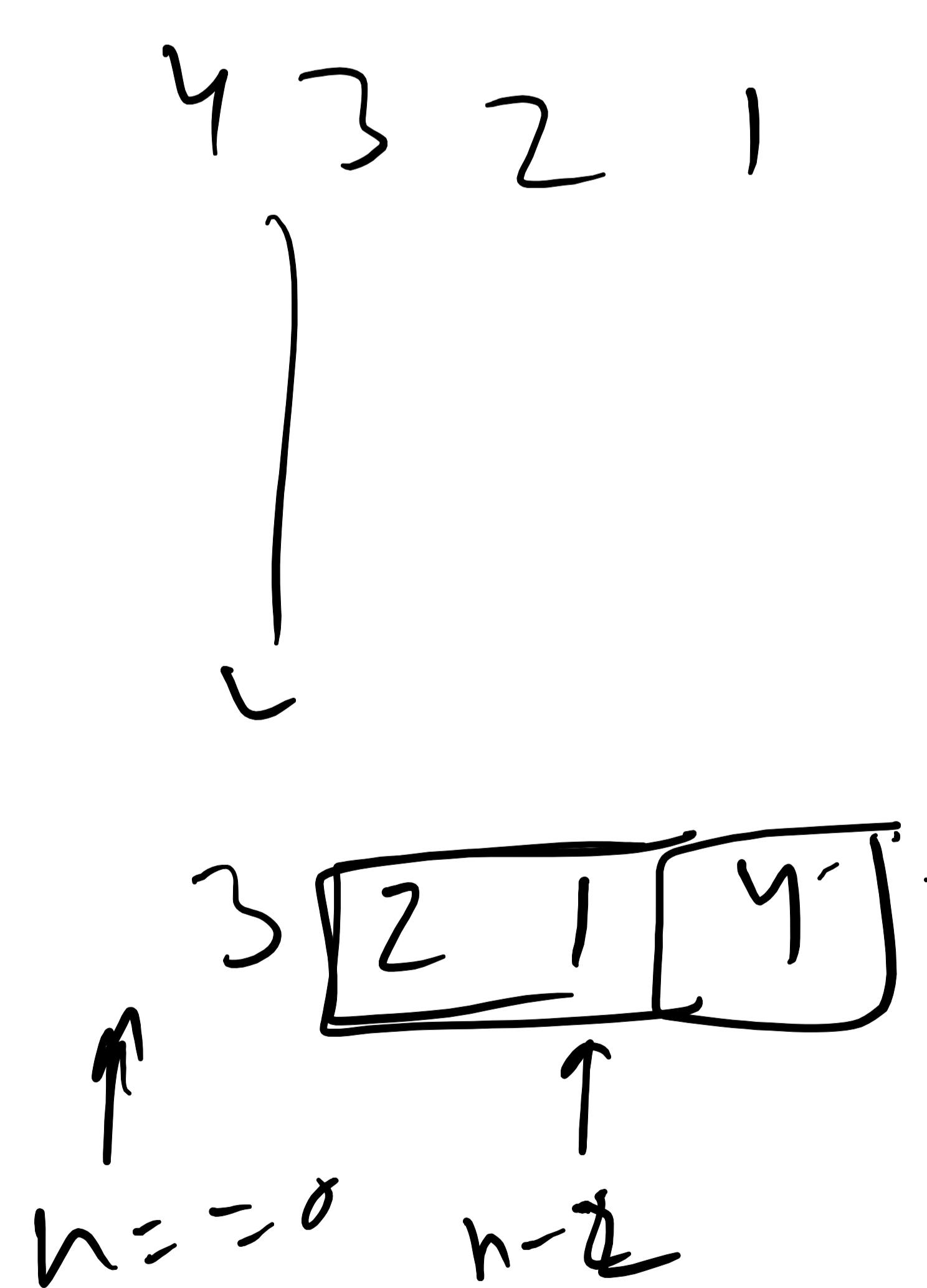
return x * power(x, y-1);

Bubble Sort

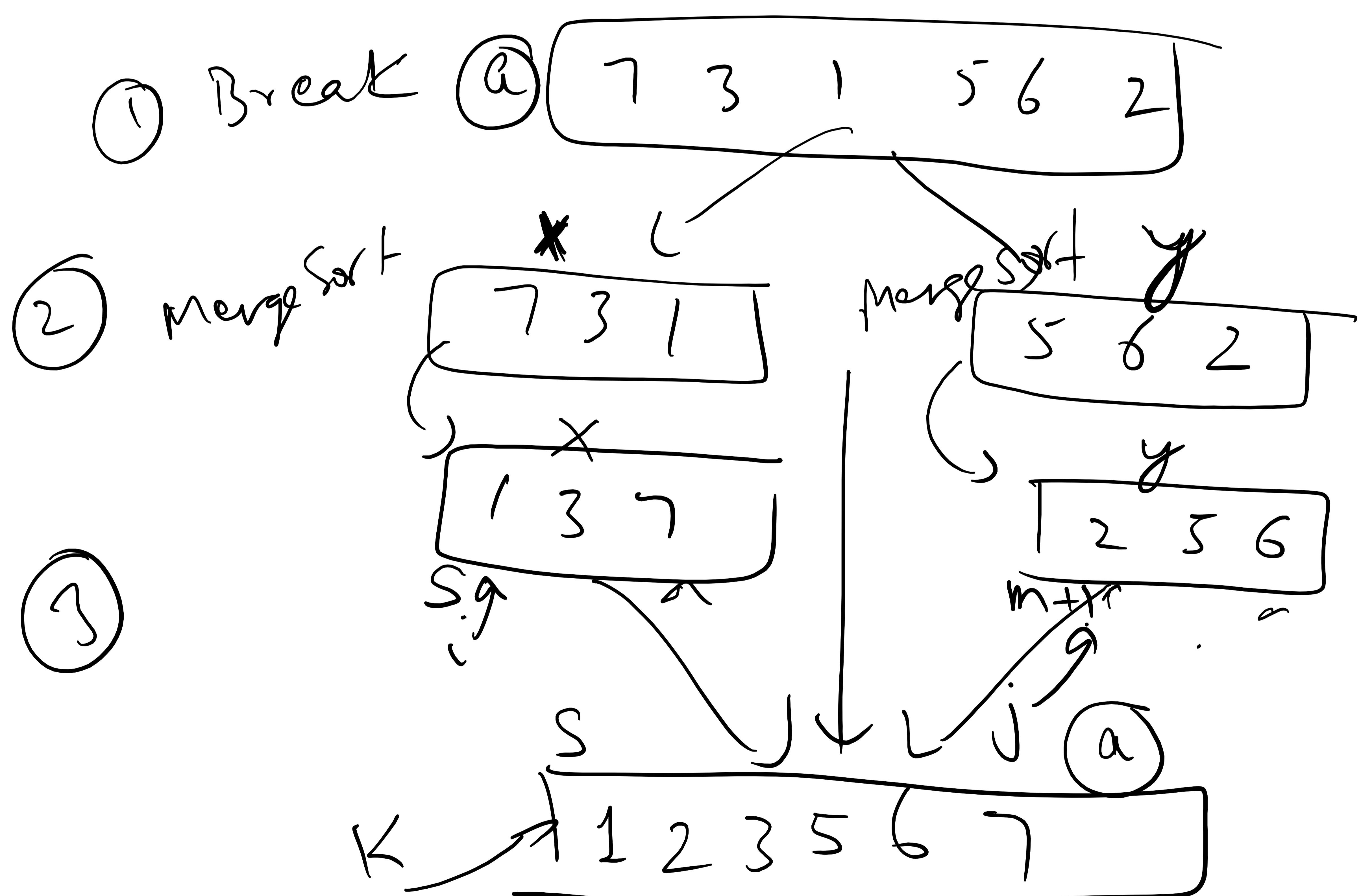
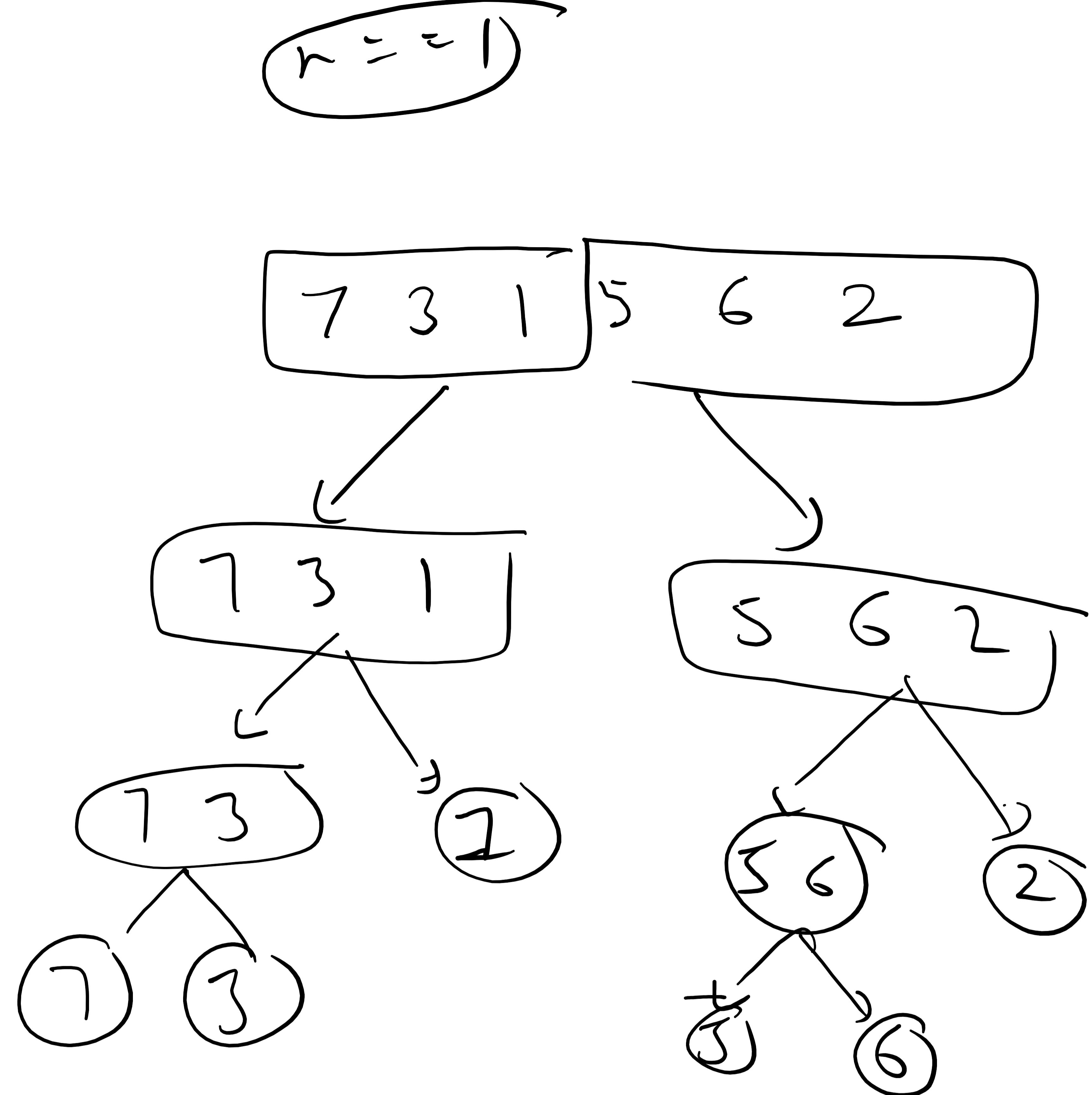
{ 4 3 2 1 } → n

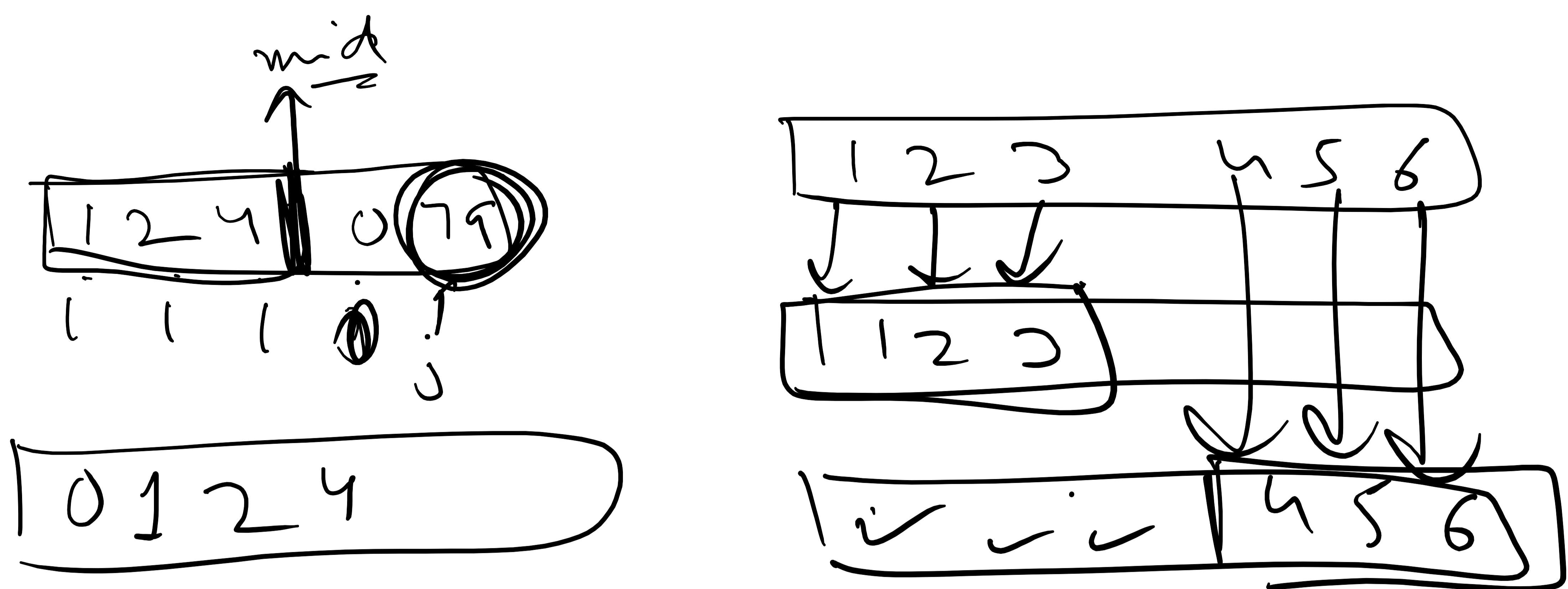
{ 4 3 2 1 } [5] → n - 1

{ 3 2 1 } [4 5] → n - 2



{ 1 2 3 4 5 }





`while (j <= end)`

$$a(k) = g(i)$$

$i++$;
 $k++$;

