① (a)

$$f2(n) = \begin{cases} 0 & n = 0 \\ n-1 & n > 0 \end{cases}$$

* All inputs are assumed non-neg.

* loop always increases the counter.

For $n=0$ case: Plug in 0 to the func f2 & check output.

For $n>0$ case:

Claim: In this case, if f1 is applied to $(0,0)$ n times, the result is $(n, n-1)$. —— $S(n)$

$n=1$: (Base) $\quad (0,0) \xrightarrow{f1} (1,0) = (n, n-1)$

So $S(1)$ is true.

Ind hyp: Suppose $S(k)$ is true for some $k \geq 1$, i.e., f1 applied to $(0,0)$ k times gives $(k, k-1)$

So, $f1^{k+1}(0,0) = f1\left(f1^{k}(0,0)\right) = f1(k, k-1)$
$= (k+1, k)$
$= (k+1, (k+1)-1)$

$\therefore S(k+1)$ is true. $\square$

Inside f2: $(n \geq 1)$ $\quad (0,0) \xrightarrow[\text{Ind}]{f1^{n}} (n, n-1) \rightsquigarrow$ final_p

final_p.snd : $n-1$ → return

f2 $f(n) \to$ time for n times looping

$f(n) = f(n-1) + \underbrace{c_1 + 3c_2 + c_3}_{c}$

$= f(n-1) + c$

$= f(n-1) + c$

$= f(0) + n \cdot c = d + c \cdot n = O(n)$

$T(n) = k + f(n) = k + O(n) = O(n)$

$O(n)$ is "like" an u.b

$\Omega(n)$ is l.b

$\Theta(n)$ is =

$T(n) \in O(n^2)$?

$T(n) \in \Omega(n^2)$?

$\boxed{\begin{array}{l} T(n) \text{ is } O(n) \\ T(n) \text{ is } \Omega(n) \\ T(n) \text{ is } \Theta(n) \end{array}}$

(b)

$$\boxed{f3(x,y) \stackrel{?}{=} x-y \qquad \times}$$

$$f3(x,y) = \begin{cases} x-y & x \geq y \\ 0 & x < y \end{cases}$$

Idea:

$x \geq y$: $(x,y) \xrightarrow{f3} (x-1, y-1) \rightarrow \cdots \rightarrow (x-y, y-y)$
$= (x-y, 0)$

$x < y$: $(x,y) \xrightarrow[\text{(else)}]{f3} (x-1, y-1) \xrightarrow[\text{else}]{} \cdots \xrightarrow[\text{else}]{} (x-x, y-x) = (0, y-x)$
$(0, y-x) \xrightarrow[\text{(else)}]{f3} (0, y-x-1) \rightarrow \cdots \rightarrow (0,0)$

if condition

when $b = f4(a, n/2)$ ⇝ sorry! Typo
$T_2(n) = T_1\left(\frac{n}{2}\right) + O(1) = O(n)$

If change $f4(a, n/2)$ to $f5(a, n/2)$ then:

$$T_2(n) = T_2(n/2) + O(1)$$
$$= a \cdot T_2(n/b) + O(n^d) \qquad (a=1,\ b=2, d=0)$$

Master thm:

$d = 0$
$\log_b a = 0$

$$T_2(n) = O(n^d \cdot \log n)$$
$$= O(\log n)$$

f4 is exponentiation in $O(n)$ time.
f5 is " " $O(\log n)$ time.
→ fast exponentiation