Análisis y Diseño de Algoritmos
Tarea 1 - Posibles respuestas

- Demuestra que $1(1+1)+2(2+1)+...+n(n+1) = \frac{n(n+1)(n+2)}{3}$ para $n \geq 0$.

  Para $n = 0$, la suma del lado izquierdo vale $0(0+1) = 0$ y la suma del lado derecho vale $\frac{0(0+1)(0+2)}{3} = 0$, por lo que coinciden. Supondremos que esto se cumple para cierta n y demostraremos por inducción que es cierto para $n+1$. Tenemos que $1(1+1)+2(2+1)+...+n(n+1)+(n+1)((n+1)+1) = \frac{n(n+1)(n+2)}{3} + (n+1)((n+1)+1)$ por la hipótesis de inducción. Buscaremos tener el mismo denominador desarrollándolo como $\frac{n(n+1)(n+2)}{3} + \frac{3(n+1)(n+2)}{3}$. Factorizamos para obtener $\frac{(n+1)(n+2)(n+3)}{3}$ que coincide con la fórmula para $n+1$.

- Demuestra que el siguiente algoritmo para calcular $n!$ es correcto:

  función FACTORIAL($n \in \mathbb{N}$)
  si $n = 0$ **entonces**
  regresa 1
  si no
  regresa $n \times$ Factorial$(n-1)$

  Para $n = 0$ el algoritmo devuelve correctamente $0! = 1$. Suponemos que el algoritmo es correcto para cierta $n-1$ con $n > 0$ y demostraremos por inducción que es correcto para n. En el caso recursivo, nuestra función devuelve $n \times$ Factorial$(n-1)$ y por la hipótesis de inducción, esto es igual a $n \times (n-1)!$ que es igual a $n!$, por lo que el algoritmo es correcto.

- Demuestra que el siguiente algoritmo para calcular $n!$ es correcto:

  función FACTORIAL($n \in \mathbb{N}$)
  $r \leftarrow 1$
  mientras $n \neq 0$
  $r \leftarrow r \times n$
  $n \leftarrow n - 1$
  regresa r

  Este algoritmo calcula factorial, pero el problema es que los factores van apareciendo como $(n)(n-1)\dots(1)$ en lugar de $(1)(2)\dots(n)$ que hubiera sido más fácil de demostrar. Sean r, n los valores iniciales de las variables correspondientes, proponemos las siguientes invariantes: $r_t = (n)(n-1)\dots(n-t+1)$ donde r_t es el valor de r en la t-ésima evaluación de la condición (comenzando a partir de 0) y $n_t = n - t$. Demostraremos las invariantes por inducción.
  Sabemos que $n_0 = n$ porque aún no se ha entrado al ciclo y la invariante $n_0 = n - 0 = n$ se cumple en este caso. Por inducción, supondremos que n_t es correcta para cierta t y demostraremos que sigue siendo correcta para $t+1$. Del código sabemos que $n_{t+1} = n_t - 1$ y por la hipótesis de inducción

esto queda $n_{t+1} = (n-t) - 1$ que es igual a $n - (t+1)$ que es el valor esperado. Entonces también sabemos que el último valor de t (cuando la condición es falsa) es $t = n$ ya que $n_n = n - n = 0$.
También demostraremos que la invariante de r_t es correcta. Tenemos que $r_t = (n)(n-1)\dots(n-t+1)$ para $t = 0$ en realidad no tiene factores, por lo que $r_0 = 1$ es correcto. Supondremos que r_t es correcta para cierta t y demostraremos que sigue siendo correcta para $t+1$. Del código sabemos que $r_{t+1} = r_t \times n_t$ y sabemos que $n_t = n-t$, por lo que $r_{t+1} = r_t \times (n-t)$. Por la hipótesis de inducción, $r_{t+1} = (n)(n-1)\dots(n-t+1) \times (n-t)$ que es igual a $(n)(n-1)\dots(n-t+1) \times (n-(t+1)+1)$, por lo que la invariante es correcta.
Finalmente, observamos que $r_n = (n)(n-1)\dots(n-n+1) = (n)(n-1)\dots(1) = n!$ que es el resultado que devuelve el algoritmo y es el que buscábamos.

- Demuestra que $5n - 10 \in \Theta(n)$. En este problema deben notar que usé Θ y no O, por lo que deben probar que $5n - 10 \in O(n)$ y $5n - 10 \in \Omega(n)$. Para demostrar que $5n - 10 \in O(n)$ pueden usar $c = 5, n_0 = 0$, por lo que ahora hay que demostrar que $5n - 10 \leq 5n$ para $n \geq 0$ que es trivial. Para demostrar que $5n - 10 \in \Omega(n)$ pueden usar $c = 4, n_0 = 10$, por lo que ahora hay que demostrar que $5n - 10 \geq 4n$ para $n \geq 10$. La expresión la pueden reescribir como $4n + n - 10 \geq 4n$ y saben que $n - 10$ es positivo porque $n \geq 10$, lo que concluye la demostración.

- Resuelve la siguiente recurrencia de forma exacta:

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

  Por un error mío, este problema se complicó terriblemente (no era mi intención poner el +1). Por sustitución repetida se ve claramente que uno de los sumandos se volverá un factorial, pero el resto queda de la forma $1 + (n) + (n)(n-1) + (n)(n-1)(n-2) + (n)(n-1)(n-2)(n-3) + \dots$ que está horrible. Si lo que quieren es encontrar una expansión compacta para esa expresión, pueden consultar `https://bit.ly/2WwoSoy`. Para nuestro problema, la expresión final es 1 cuando $n = 0$ y $\lfloor e \times n! \rfloor$ cuando $n > 0$. La pondré como correcta si al menos hicieron sustitución repetida y apareció el patrón. Pueden consultar un programa que compara diferentes formas de realizar el cálculo en `https://bit.ly/2YYFOS8`.

- Encuentra la complejidad asintótica de la siguiente recurrencia usando el teorema maestro:

$$T(n) = \begin{cases} 0 & n = 0 \\ 2T(\frac{n}{2}) + n\,log_2(n) & n > 0 \end{cases}$$

  En esta recurrencia aplica el caso del teorema maestro donde $f(n) = O(n^k log_2(n)^h)$ y se cumple que $k = log_b a = log_2 2 = 1$, por lo que la complejidad de $T(n)$ es $O(n\,(log_2(n))^2)$.