

Gael Balderrama Dominguez  
Tarea-06  
Noviembre 2023

Problema 4

La llamada (bundle '("a" "b" "c") 0) es un buen uso de bundle ¿qué produce? ¿por qué?

Respuesta:

Es un buen uso, ya que deja ver como la funcion maneja los casos donde el numero en el que se van a separar las listas unitarias es menor que el numero de listas y tambien que se hace cuando la lista es 0, dando como resultado, en este caso, el resultado seria regresar la lista vacia con la definicion de bundle que manejamos

Problema 10

Si la entrada a quicksort contiene varias repeticiones de un numero, va a regresar una lista estrictamente más corta que la entrada. Responde el por qué y arregla el problema.

Respuesta:

Cuando un elemento es igual al pivote, no son incluidos a la lista resultante, si no nomas se agrega el pivote, dando esto como resultado el que la lista resultante sea mas corta que la lista original ya que elimina los elementos duplicados

Problema 18

¿En qué casos el subproblema no es estructuralmente más pequeño que el problema original en la definición de la función de Ackermann?

Respuesta:

El caso en el que el subproblema no es mas pequeño que el problema original es en la tercera linea de la definicion de la funcion de Ackermann, ya que  $m$  y  $n$  se reducen pero tambien se utiliza  $m$  sin reducir, haciendo que el problema no disminuya en cada llamada recursiva

Problema 19

Describe con tus propias palabras cómo funciona find-largestdivisor de gcd-structural. Responde por qué comienza desde (min n m).

Respuesta:

La funcion find-largestdivisor en gcd-structural funciona con un condicional con 3 posibles casos, uno donde el valor con el que se llama find-largestdivisor es igual a 1 entonces se regresa 1, otro donde se revisa si el valor del residuo de  $n$  y  $k$ , y el residuo de  $m$  y  $k$ , son iguales, si son iguales se regresa  $k$ , y en otro caso, se vuelve a llamar a find-largestdivisor con el valor de  $k-1$  El porque se elige el minimo entre  $n$  y  $m$  es que solo es necesario comprobar los divisores menores o iguales al numero menor entre los 2 ya que si uno es menor que otro entonces es divisor del mas grande

Problema 20

Describe con tus propias palabras cómo funciona find-largestdivisor de gcd-structural. Responde por qué comienza desde (min n m).

Respuesta:

El `find-largestdivisor` en `gcd-generative` recibe 2 argumentos, el *max* y *min* que representan los numeros mas grandes que se comparan para el Maximo comun divisor, si el *min* es igual a cero se devuelve *max* ya que seria el maximo comun divisor de los 2 numeros, si este no es el caso se llama recursivamente a `find-largestdivisor` con *min* y el residuo del *max* y el *min*

#### Problema 22

Piensa y Describe por qué no siempre es la mejor opción elegir el procedimiento más eficiente en tiempo de ejecución. Utiliza criterios que no sean el de "Eficiencia" Respuesta:

Uno de los factores que podria llegar a ser mejor que la eficiencia del procedimiento seria la facilidad de compresion y el uso del mismo, si se tiene un procedimiento que es muy eficiente pero es dificil de comprender no se podria utilizar de manera tan efectiva o tan flexible en diferentes casos, haciendo que su uso se vea limitado a casos especificos.