

Universidad Nacional de Colombia

Facultad de Ingeniería

Computación Paralela y Distribuida

Profesor: Oscar Agudelo R.

Taller post Semana Santa - Recursividad

Nombre del estudiante: Gabriel Andres Anzola Tachak

1. Responder las 13 preguntas prácticas planteadas al final del archivo Que_es_la_recursividad_2024.pdf.

1. En general, ¿qué es una cosa recursiva?

Una cosa recursiva es algo cuya definición se incluye a sí misma. Es decir, tiene una definición autorreferencial.

2. En programación, ¿qué es una función recursiva?

Es una función que se llama a sí misma.

3. ¿Cuáles son las cuatro características que tienen las funciones?

- Las funciones tienen código que es ejecutado cuando se invoca la función.
- Los argumentos se pasan a la función cuando se invoca. Esta es la entrada a la función, y las funciones pueden tener cero o más argumentos de entrada.
- Las funciones retornan un valor. Esta es la salida de la función, aunque algunos lenguajes de programación permiten que las funciones no devuelvan nada o que devuelvan valores nulos como undefined o None.
- El programa recuerda desde qué línea de código se llamó a la función y vuelve a dicha línea cuando la función termina su

ejecución.

4. ¿Qué es una pila?

Una pila es una de las estructuras de datos más sencillas de la informática. Almacena múltiples valores como una lista, pero a diferencia de las listas, sólo permite añadir o eliminar valores de la "parte superior" de la pila. En las pilas implementadas con listas o matrices, el "tope" es el último elemento, en el extremo derecho de la lista o matriz.

5. ¿Cuáles son los términos para añadir y eliminar valores de la parte superior de una pila?

Añadir valores se llama empujar (**pushing**) valores a la pila, mientras que eliminar valores se llama sacar (**popping**) valores de la pila.

6. Supongamos que pone la letra J en una pila, luego coloca la letra Q, luego retira de la pila, luego coloca la letra K, luego vuelve a retirar de la pila. ¿Qué aspecto tiene ahora la pila?

La pila luciría así, con un solo elemento (J) en la parte superior de la pila:

Pila

J (*tope de la pila*)

7. ¿Qué se coloca y se extrae en la pila de llamadas?

Los objetos Frame, que contienen información sobre una única llamada a una función, incluyendo qué línea de código llamó a la función, de modo que la ejecución puede volver a dicha línea de código cuando la función retorna. Son añadidos a la pila de llamadas cuando se llama a una función y son eliminados de la pila de llamadas cuando la función retorna.

8. ¿Qué provoca un desbordamiento de la pila?

Las constantes llamadas a funciones sin retorno hacen crecer la pila de llamadas hasta agotar toda la memoria del ordenador asignada a la pila de llamadas. Esto se conoce como desbordamiento de la pila.

9. ¿Qué es un caso base?

Es un caso o conjunto de circunstancias en el que la función deja de llamarse a sí misma y en su lugar simplemente retorna.

10. ¿Qué es un caso recursivo?

Es un caso en el que la función se llama a sí misma recursivamente.

11. ¿Cuántos casos base y recursivos tienen las funciones recursivas?

Al menos un caso base y al menos un caso recursivo.

12. ¿Qué ocurre si una función recursiva tiene cero casos base?

La función nunca deja de hacer llamadas recursivas y acaba provocando un desbordamiento de la pila.

13. ¿Qué pasa si una función recursiva tiene cero casos recursivos?

La función nunca se llama a sí misma y esta sería una función ordinaria más no una función recursiva.

2. ¿Qué se explica en el intervalo de 6:00 a 6:20 del video "!!Con 2019- Tail Call Optimization: The Musical!! by Anjana Vakil & Natalia Margolis"? Haga un breve resumen de lo explicado en ese intervalo.

Tail Call Optimization (TCO) como una técnica en funciones recursivas para evitar los desbordamientos de pila (Stack Overflow). Se explica como los frames del medio de la pila de llamadas son eliminados para evitar el desbordamiento de la pila ya que no

tienen trabajo pendiente por hacer y le retornan el resultado al siguiente frame en la pila de llamadas.

3. Responder la 6 preguntas prácticas planteadas al final del archivo Tail_Call_Optimization_2024.pdf.

1. ¿Qué evita la optimización de llamadas de cola?

La optimización de llamadas de cola evita los desbordamientos de pila, eliminando la necesidad de retener los objetos Frame en la pila de llamadas cuando la última acción de una función recursiva es devolver el valor de una llamada recursiva.

2. ¿Qué tiene que hacer la última acción de una función recursiva para que la función pueda ser recursiva de cola?

La última acción de una función recursiva debe ser devolver el valor de la llamada recursiva. No puede haber ninguna instrucción entre la llamada recursiva y la sentencia return para que sea recursiva de cola.

3. ¿Todos los compiladores e intérpretes implementan la optimización de llamadas de cola?

No todos los compiladores e intérpretes implementan la optimización de llamadas de cola. Por ejemplo, CPython (el intérprete de Python) y el compilador de Java a partir de la versión 8 no soportan esta optimización, mientras que Safari es el único navegador que la implementa para JavaScript.

4. ¿Qué es un acumulador?

Un acumulador es un parámetro que actúa como una solución en la recursividad de cola para almacenar un resultado parcial de un cálculo, que de otro modo se habría almacenado en una variable local. Permite que la función recursiva devuelva el resultado acumulado cuando se alcanza el caso base.

5. ¿Cuál es la desventaja de la recursividad de cola?

La desventaja de la recursividad de cola es que requiere reorganizar el código de la función, lo que puede hacerlo más difícil de comprender y menos legible. Además, la optimización de llamadas de cola no está disponible en todos los lenguajes o intérpretes.

6. ¿Se puede reescribir el algoritmo quicksort (tratado en el capítulo 5 del libro) para utilizar la optimización de llamadas de cola?

No se puede reescribir el algoritmo quicksort para usar la optimización de llamadas de cola, porque este algoritmo requiere dos llamadas recursivas en la mayoría de las implementaciones. Como la recursividad de cola solo puede aplicarse cuando la última acción es devolver el valor de una llamada recursiva, esto no es posible en quicksort, ya que al menos una de las dos llamadas no puede ser la última acción.
