Computación y estructuras discretas II

Unidad Programación Funcional: Recursividad

Juan Marcos Caicedo Mejía – jmcaicedo@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes
Facultad Barberi de Ingeniería, Diseño y Ciencias Aplicadas
Universidad ICESI

Tabla de contenido

Contenido

Recursividad

Definición

Recursión es el proceso definir objetos en términos de ellos mismos.

Recursividad

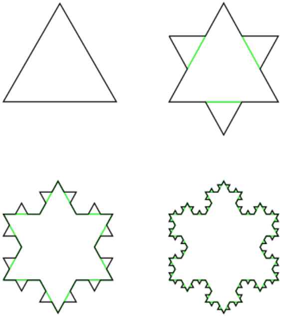
Definición

Recursión es el proceso definir objetos en términos de ellos mismos. Considere el fractal Koch de copo de nieve.

Recursividad

Definición

Recursión es el proceso definir objetos en términos de ellos mismos. Considere el fractal Koch de copo de nieve.



Definición recursiva del fractal de Koch

Alteramos recursivamente cada segmento de línea del fractal de la siguiente manera:

Definición recursiva del fractal de Koch

Alteramos recursivamente cada segmento de línea del fractal de la siguiente manera:

1. Divida el segmento de línea en tres segmentos de igual longitud.

Definición recursiva del fractal de Koch

Alteramos recursivamente cada segmento de línea del fractal de la siguiente manera:

1. Divida el segmento de línea en tres segmentos de igual longitud.
2. Dibuje un triángulo equilátero que tenga el segmento medio del paso 1 como base y apunte hacia afuera.

Definición recursiva del fractal de Koch

Alteramos recursivamente cada segmento de línea del fractal de la siguiente manera:

1. Divida el segmento de línea en tres segmentos de igual longitud.
2. Dibuje un triángulo equilátero que tenga el segmento medio del paso 1 como base y apunte hacia afuera.
3. Elimine el segmento de línea que es la base del triángulo del paso 2.

Recursividad

Definición

Una función es recursiva si se llama a sí misma.

Recursividad

Definición

Una función es recursiva si se llama a sí misma.

Debe tener:

- Caso base
- Caso recursivo

Recursividad

¿Cómo se define una función de manera recursiva?

Para definir una función, cuyo dominio es el conjunto de los enteros no negativos, de manera recursiva, utilizamos dos pasos:

Paso base: Se especifica el valor de la función en un valor inicial.

Paso recursivo: Se proporciona una regla para obtener su valor en un entero utilizando valores enteros más pequeños.

Ejemplo

Suponga que f se define recursivamente como:

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Solución

A partir de la definición recursiva se obtiene:

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Solución

A partir de la definición recursiva se obtiene:

$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Solución

A partir de la definición recursiva se obtiene:

$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Solución

A partir de la definición recursiva se obtiene:

$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$

$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$

Recursividad

Ejemplo

Suponga que f se define recursivamente como:

Paso base: $f(0) = 3$

Paso recursivo: $f(n+1) = 2f(n) + 3$

Obtenga $f(1)$, $f(2)$, $f(3)$ y $f(4)$.

Solución

A partir de la definición recursiva se obtiene:

$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$

$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$

$f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$

Recursividad

Ejemplo

Dé una función recursiva de la función factorial $F(n) = n!$.

Recursividad

Ejemplo

Dé una función recursiva de la función factorial $F(n) = n!$.

Solución

A partir de la definición recursiva se obtiene:

Ejemplo

Dé una función recursiva de la función factorial $F(n) = n!$.

Solución

A partir de la definición recursiva se obtiene:

Paso base: $F(0) = 1$

Recursividad

Ejemplo

Dé una función recursiva de la función factorial $F(n) = n!$.

Solución

A partir de la definición recursiva se obtiene:

Paso base: $F(0) = 1$

Paso recursivo: $F(n+1) = (n+1)F(n)$

Las definiciones recursivas de conjuntos también tienen dos partes.

Paso base: Se especifica una colección inicial de elementos.

Paso recursivo: Se proporciona una regla para la formación de nuevos elementos del conjunto a partir de los que ya se conocen.

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Solución

Los nuevos elementos de S se van construyendo a partir del paso base de la siguiente manera:

- $3 \in S$ luego $S = \{3\}$

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Solución

Los nuevos elementos de S se van construyendo a partir del paso base de la siguiente manera:

- $3 \in S$ luego $S = \{3\}$
- $3 \in S$ luego $3 + 3 = 6 \in S$ luego $S = \{3, 6\}$

Describa el conjunto S por comprensión.

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Solución

Los nuevos elementos de S se van construyendo a partir del paso base de la siguiente manera:

- $3 \in S$ luego $S = \{3\}$
- $3 \in S$ luego $3 + 3 = 6 \in S$ luego $S = \{3, 6\}$
- $3, 6 \in S$ luego $3 + 6 = 9 \in S$ luego $S = \{3, 6, 9\}$

Describa el conjunto S por comprensión.

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Solución

Los nuevos elementos de S se van construyendo a partir del paso base de la siguiente manera:

- $3 \in S$ luego $S = \{3\}$
- $3 \in S$ luego $3 + 3 = 6 \in S$ luego $S = \{3, 6\}$
- $3, 6 \in S$ luego $3 + 6 = 9 \in S$ luego $S = \{3, 6, 9\}$
- $3, 6, 9 \in S$ luego $3 + 9 = 12 \in S$ luego $S = \{3, 6, 9, 12\}$

Describa el conjunto S por comprensión.

Conjuntos definidos recursivamente

Ejemplo

Considere el subconjunto S de los enteros definido por:

Paso base: $3 \in S$

Paso recursivo: Si $x \in S$ y $y \in S$, entonces $x + y \in S$

Solución

Los nuevos elementos de S se van construyendo a partir del paso base de la siguiente manera:

- $3 \in S$ luego $S = \{3\}$
- $3 \in S$ luego $3 + 3 = 6 \in S$ luego $S = \{3, 6\}$
- $3, 6 \in S$ luego $3 + 6 = 9 \in S$ luego $S = \{3, 6, 9\}$
- $3, 6, 9 \in S$ luego $3 + 9 = 12 \in S$ luego $S = \{3, 6, 9, 12\}$
- $S = \{3, 6, 9, 12, 15, 18, 21, \dots\}$

Describa el conjunto S por comprensión.

Contenido

Ejercicios

Ejercicio

Obtenga $f(2), f(3), f(4)$ y $f(5)$ si f se define recursivamente por $f(0) = f(1) = 1$ y para $n = 1, 2, \ldots,$ como:

a) $f(n+1) = f(n) - f(n-1)$

b) $f(n+1) = f(n)f(n-1)$

c) $f(n+1) = f(n)^2 + f(n-1)^3$

d) $f(n+1) = \dfrac{f(n)}{f(n-1)}$

Contenido

Algoritmos recursivos

Definición

Un algoritmo se llama recursivo si resuelve un problema reduciéndolo a un caso del mismo problema con datos de entrada más pequeños.

Algoritmos recursivos

Definición

Un algoritmo se llama recursivo si resuelve un problema reduciéndolo a un caso del mismo problema con datos de entrada más pequeños.

Este tipo de algoritmos hallan la solución al problema mediante una secuencia de reducciones, hasta que se llega a un caso inicial cuya solución se conoce.

Algoritmos recursivos

Ejemplo

Dé un algoritmo recursivo para hallar a^n, donde a es un número real distinto de cero y n un entero no negativo.

Algoritmos recursivos

Ejemplo

Diseñe un algoritmo recursivo que calcule el factorial.

Algoritmos recursivos

Ejemplo

Diseñe un algoritmo recursivo que calcule el factorial.

Solución

```
procedure factorial(n : entero positivo)
    if n = 1 then
        factorial(n) := 1
    else
        factorial(n) := n * factorial(n - 1)
```

Ejercicios

Ejercicio

Escriba un algoritmo recursivo que calcule el término n-ésimo de la sucesión definida por:

$$a_0 = 1, \quad a_1 = 2, \quad a_n = a_{n-1} \cdot a_{n-2}, \quad \text{para } n = 2, 3, 4, \ldots$$

Ejercicios

Ejercicio

Escriba un algoritmo recursivo que calcule el término n-ésimo de la sucesión definida por:

$$a_0 = 1, \quad a_1 = 2, \quad a_2 = 3,$$

$$a_n = a_{n-1} + a_{n-2} + a_{n-3}, \quad \text{para } n = 3, 4, 5, \ldots$$

Recursividad: Factorial

```
def factorial(n: Int): Int = {
  if (n == 0) 1
  else n * factorial(n - 1)
}
```

Recursividad: Factorial

```
def factorial(n: Int): Int = {
  if (n == 0) 1
  else n * factorial(n - 1)
}
```

Caso base: n == 0 Caso recursivo: n * factorial(n - 1)

Recursividad: Fibonacci

```
def fibonacci(n: Int): Int = {
  if (n <= 1) n
  else fibonacci(n - 1) + fibonacci(n - 2)
}
```

Recursividad: Fibonacci

```
def fibonacci(n: Int): Int = {
  if (n <= 1) n
  else fibonacci(n - 1) + fibonacci(n - 2)
}
```

Nota: esta versión no es eficiente.