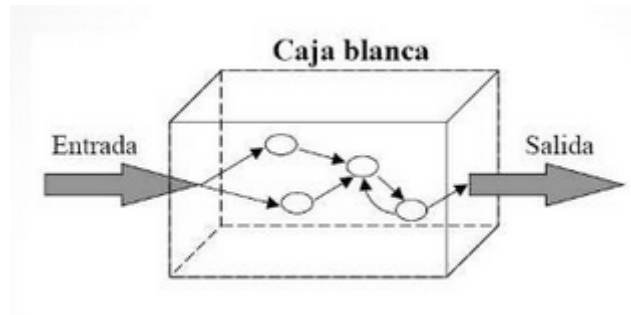


Apuntes. Prueba del camino básico

Introducción

La prueba del camino básico, es una prueba de **caja blanca** que consiste en verificar que todas las instrucciones del programa se ejecutan por lo menos una vez.



Esta técnica permite obtener una medida de la **complejidad lógica** de un bloque de código o de una función y usar esa medida como guía para la definición de un conjunto básico (diseño de casos de prueba) de **caminos de ejecución**.

Los casos de prueba que obtengamos garantizarán que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Pasos

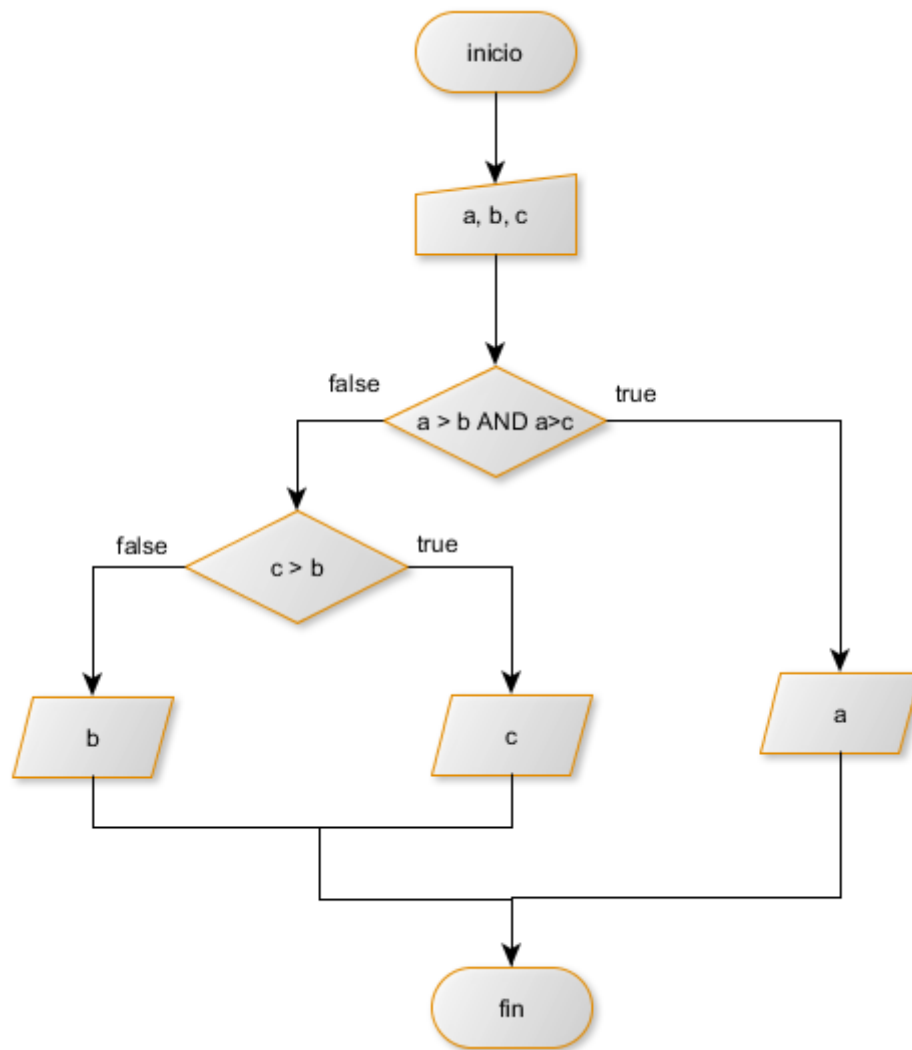
1. Programa al que queremos realizar pruebas

Partimos de un bloque de código o función al que queremos elaborar casos de prueba usando esta técnica

Por ejemplo una función a la que le pasamos tres valores y no muestra cuál es el mayor.

```
def mayor_tres(a, b, c):  
    if a > b and a > c:  
        return a  
    if c > b:  
        return c  
    else:  
        return b
```

2. Dibujamos el diagrama de flujo del programa

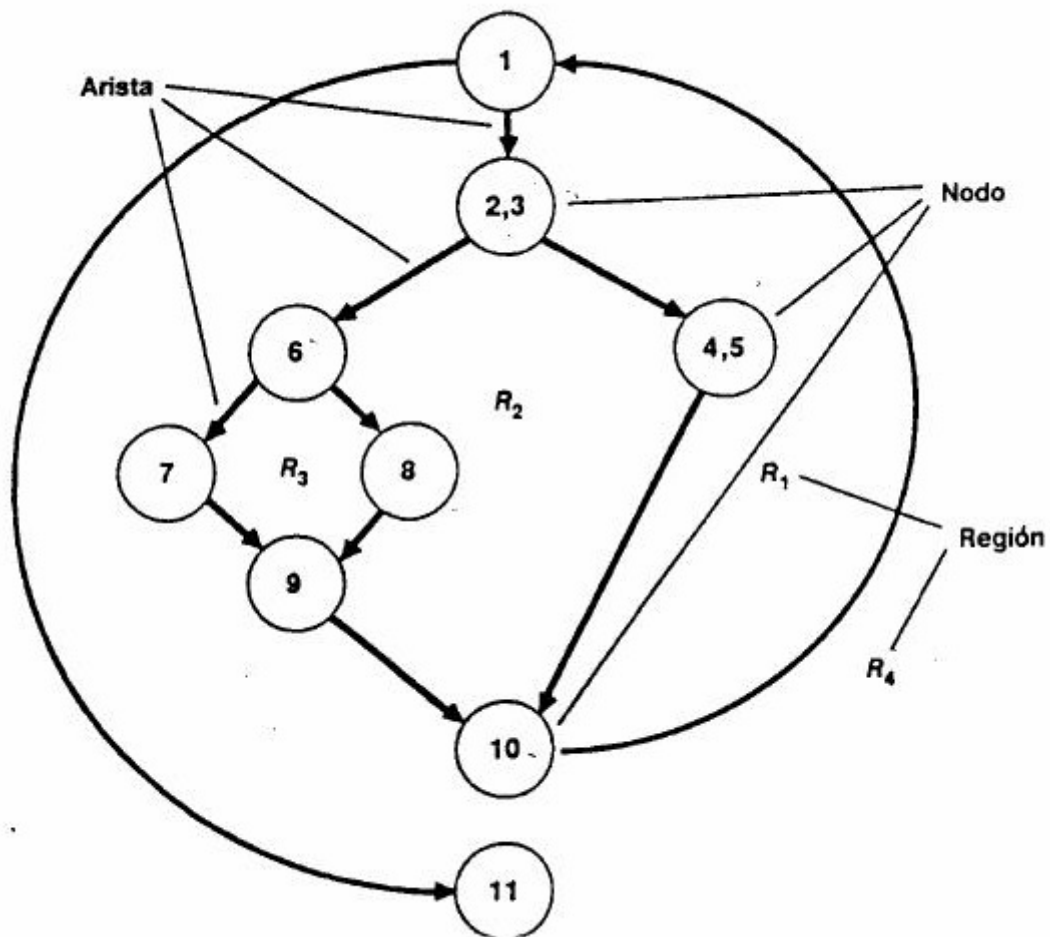


3. Dibujar el **grafo de flujo**

Un grafo de flujo es un diagrama que nos permite representa el flujo de control lógico de un algoritmo.

Los elementos de un grafo de flujo son:

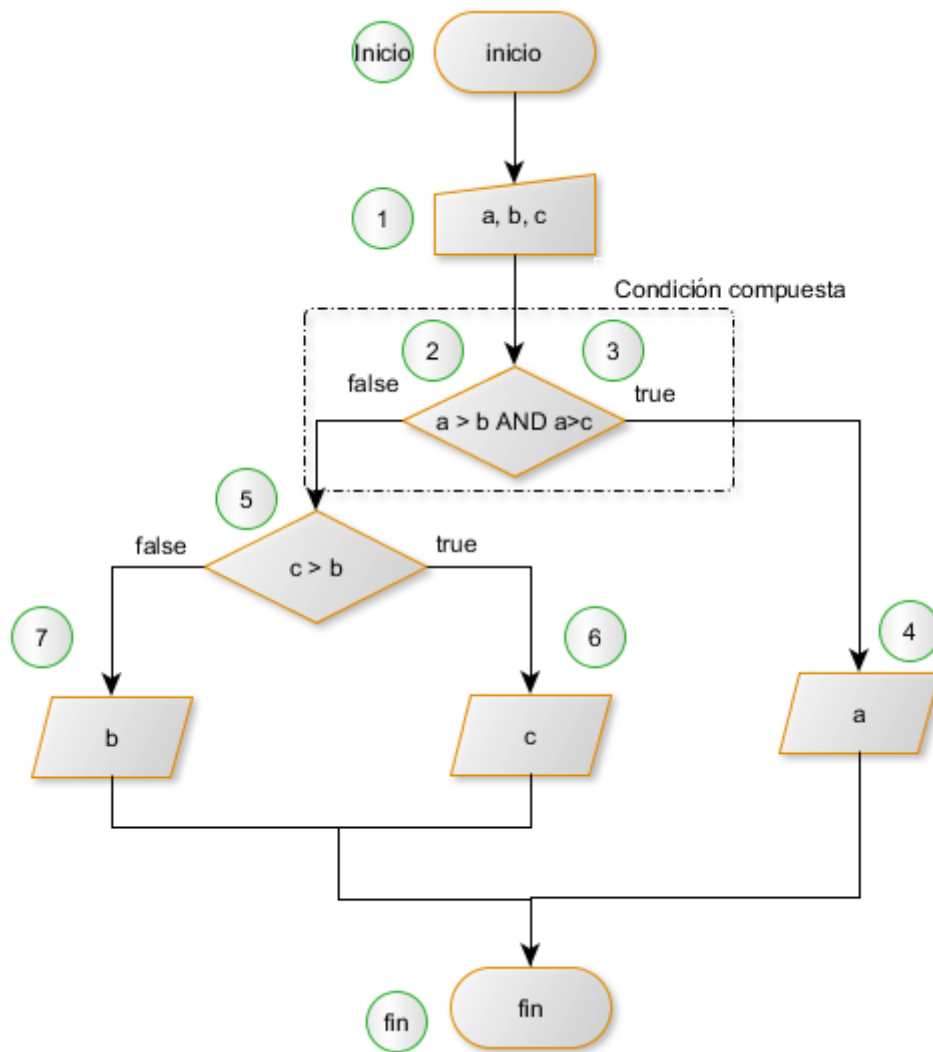
- **Nodos:** representa una o más sentencias procedimentales.
- **Aristas** (flechas) representan el flujo de control.
- **Nodos predicado:** nodos que contienen una **condición**
- **Regiones:** áreas delimitadas por las aristas y nodos.



Para elaborar un grafo de flujo partimos del diagrama de flujo del programa y seguimos las siguiente reglas:

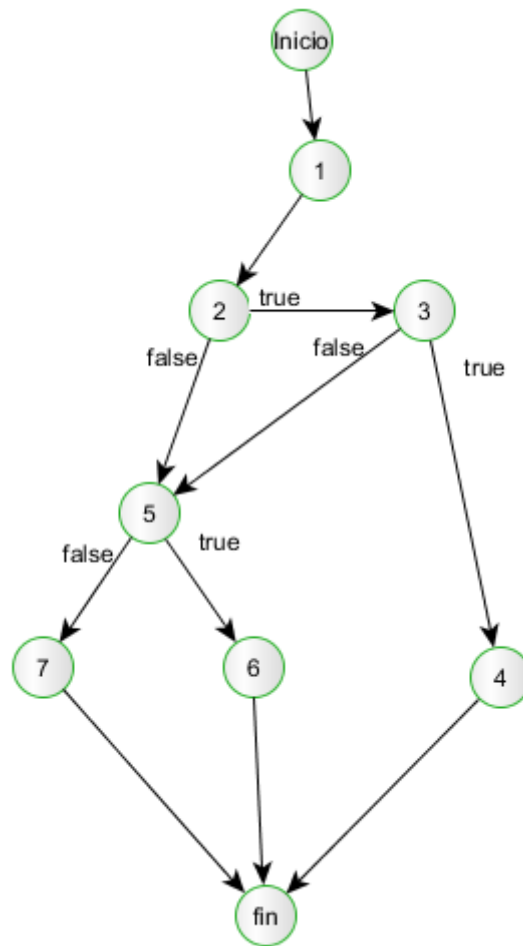
- Si en el digrama de flujo hay una **secuencia de pasos** se pueden agrupar en **un solo nodo**.
- A los nodos se les asigna un **identificador único**
- Si en el diagrama de flujo el código se **bifurca** se genera una **arista por cada posible salida**. Dichas aristas se pueden **etiquetar** para facilitar la comprensión del grafo.
- Si en el diagrama de flujo se utilizan **condiciones compuestas**, la generación del grafo de flujo tiene que **descomponer** las condiciones compuestas en condiciones sencillas

A partir del diagrama de flujo anterior empezamos a construir el **grafo de flujo** identificando los nodos:



Fíjate que cómo tenemos una condición compuesta ($a > b \text{ and } a > c$) debemos descomponerla creando un nodo para cada una de las condiciones.

A continuación dibujamos el grafo de flujo.



4. Cálculo de la complejidad ciclomática

En un grafo de flujo la complejidad ciclomática **V(G)** indica el número máximo de caminos independientes del grafo.

Puede calcularse de tres formas alternativas:

- $V(G) = R$. Donde:
 - R es el **número de regiones** en que el grafo divide el plano.
- $V(G) = A - N + 2$. Donde:
 - A - Número de aristas
 - N - Número de nodos.
- $V(G) = P + 1$.
 - Donde P es el número de nodos predicado

Para nuestro ejemplo

- $V(G) = R = 4$ - regiones en el grafo. La región exterior al grafo también se cuenta.
- $V(G) = A - N + 2 = 11 - 9 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

Como hemos comentado, la complejidad ciclomática nos da una medida cualitativa de la complejidad **lógica** del programa. En función del valor de **V(G)** existen unos valores de referencia para establecer la complejidad lógica:

V(G)	Complejidad lógica. Riesgo de tener errores
1 - 10	Programas sencillos. Sin mucho riesgo
11 - 20	Programas más complejos. Riesgo moderado
21 - 50	Programas complejos. Alto riesgo
> 50	Programas no testeables. Riesgo muy alto

5. Determinar casos de prueba

A partir del cálculo de la complejidad ciclomática obtenemos el número de casos de prueba, esto es, el número de **caminos independientes** del grafo.

Un camino independiente es cualquier camino del programa que incluye, al menos, una arista que no haya sido utilizada por el resto de caminos independientes del programa.

Para nuestro ejemplo, de mayor a menor longitud, 4 caminos independientes pueden ser:

- Camino 1: I-1-2-3-5-6-F
- Camino 2: I-1-2-3-5-7-F
- Camino 3: I-1-2-3-4-F
- Camino 4: I-1-2-5-6-F

Observando el grafo de flujo elaboramos tabla con los caminos independientes posibles y para cada uno de ellos generamos una tabla en el que para cada camino incluimos:

- **Entrada:** condiciones de los valores de entrada que posibilitan dicho camino
- **Prueba:** valores de ejemplo que cumplen las condiciones de entrada
- **Salida:** resultado esperado a partir de los valores de prueba

Para nuestro ejemplo:

Los nodos predicados y sus condiciones son:

- **2** - a > b
- **3** - a > c
- **5** - c > b

Con los caminos y las condiciones de los nodos predicados ya podemos elaborar la tabla con los casos de prueba

Camino	Entrada	Prueba	Salida
I-1-2-3-5-6-F	a>b=True a>c=False c>b=True	a=5 b=3 c=6	6
I-1-2-3-5-7-F	a>b=True a>c=False c>b=False	a=5 b=3 c=2	5
I-1-2-3-4-F	a>b=True a>c=True	a=7 b=5 c=4	7
I-1-2-5-6-F	a>b=False c>b=True	a=5 b=6 c=8	8

6. Realización de pruebas

El paso siguiente sería utilizar alguna herramienta de realización de tests del lenguaje de programación en el que hayamos codificado nuestro programa.

Recursos

- [Prueba del camino básico - JC Mouse. Código colectivo](#)
- [Técnicas de prueba - ISIII](#)