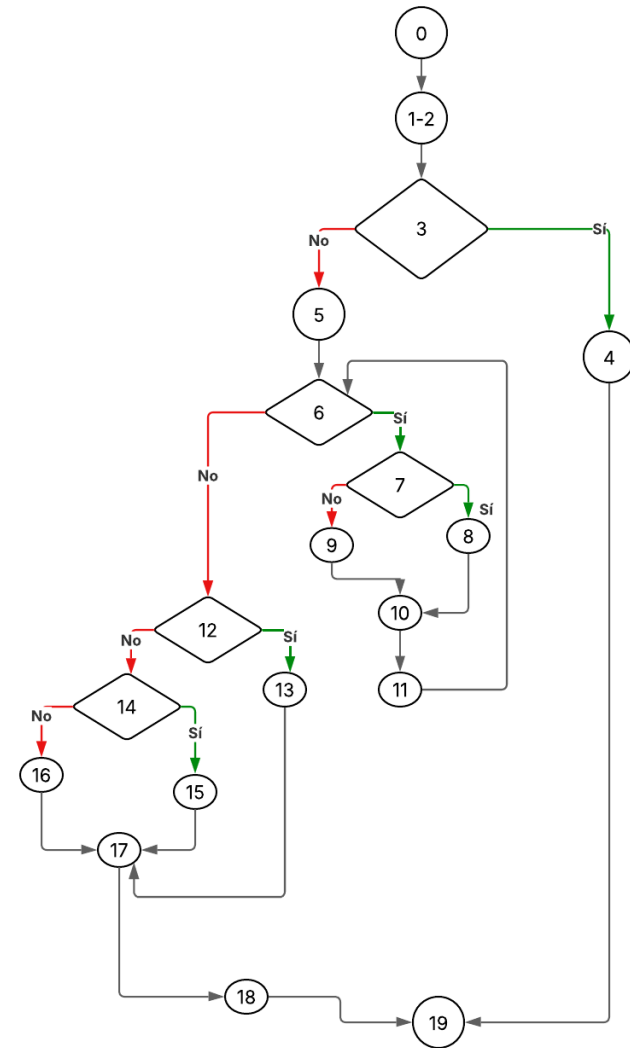


```

0: public static String cuentaPares(int[] numeros) {
1:   int contadorPares=0,contadorImpares=0;
2:   String resultado = "";
3:   if (numeros.length == 0) {
4:     return "No hay numeros";
5:   }
6:   for (int numero : numeros) {
7:     if (numero % 2 == 0) {
8:       contadorPares = contadorPares + 1;
9:     } else {
10:      contadorImpares = contadorImpares + 1;
11:    }
12:   }
13:   if (contadorPares > contadorImpares) {
14:     resultado = ("Hay más números pares: " +
15:                contadorPares);
16:   } else if (contadorImpares > contadorPares) {
17:     resultado = ("Hay más números impares: " +
18:                contadorImpares);
19:   } else {
20:     resultado=("Cantidad igual pares e impares: " +
21:              contadorPares);
22:   }
23:   return resultado;
24: }

```

## GRAFO DE FLUJO



Aplicando la fórmula, calcula aquí la complejidad del grafo  $V(G)$ :  
Nodos predicado + 1

$$5 + 1 = 6$$

Indica los nodos predicado:

$$3 - 6 - 7 - 12 - 14$$

Identifica a continuación los caminos posibles:

CAMINO	NODOS
A	0-1-2-3-4-19
B	0-1-2-3-5-6-7-8-10-11-12-13-17-18-19
C	0-1-2-3-5-6-7-9-10-11-12-14-15-17-18-19
D	0-1-2-3-5-6-7-8-9-10-11-12-13-17-18-19
E	0-1-2-3-5-6-7-8-9-10-11-12-14-15-17-18-19
F	0-1-2-3-5-6-7-8-9-10-11-12-14-16-17-18-19

Rellena a continuación la tabla de casos de prueba, justificando debajo la elección de los valores:

CAMINO	CASO DE PRUEBA	JUSTIFICACIÓN
A	numeros = {}	El array es vacío
B	numeros = {2}	El array tiene un n° par
C	numeros = {5}	El array tiene un n° impar
D	numeros = {2,4,6,5,3}	El array tiene mas n° pares
E	numeros = {1,3,5,2,4}	El array tiene mas n° impares
F	numeros = {1,3,2,4}	El array tiene igual n° de pares e impares