

# EJERCICIOS CAJA BLANCA

---

Técnica del camino básico. Ejercicios 1 y 2

# Ejercicio 1

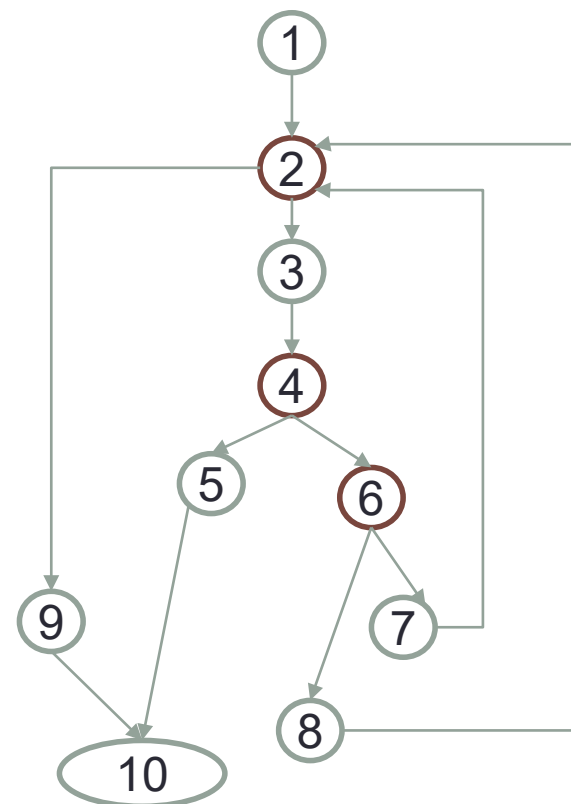
```
• static public int search(char c, char []v)
• {
•     int a, z, m;
•     a = 0;
•     z = v.Length - 1;
•     while (a <= z)
•     {
•         m = (a + z) / 2;
•         if (v[m] == c) {
•             return 1;
•         }
•         else if(v[m] < c)
•         {
•             a = m + 1;
•         }
•         else
•         {
•             z = m - 1;
•         }
•     }
•     return 0;
• }
```

# Ejercicio 1

```

• static public int search(char c, char []v)
• {
•     int a, z, m;
•     ① a = 0;
•     z = v.Length - 1;
•     ② while (a <= z)
•     {
•         m = (a + z) / 2; ③
•         ④ if (v[m] == c) {
•             return 1; ⑤
•         }
•         ⑥ else if (v[m] < c)
•         {
•             a = m + 1; ⑦
•         }
•         else
•         {
•             z = m - 1; ⑧
•         }
•     }
•     return 0; ⑨
• ⑩ }

```



$V(G) = 4$

Áreas = 4

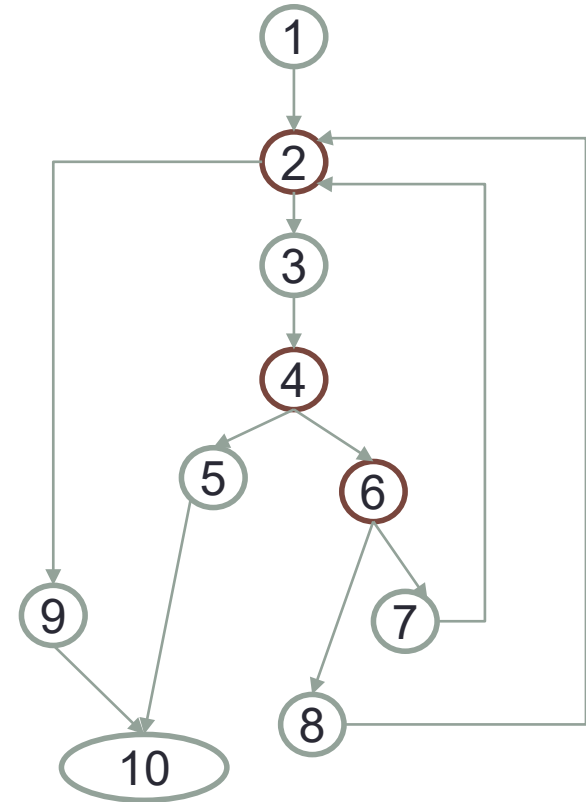
Nodos Predicado = 3  $\rightarrow 3+1=4$

Nodos = 10  $\rightarrow 12-10+2=4$

Aristas = 12

# Ejercicio 1

Path	Input	Output
{1,2,9,10} Cadena vacia	V="" c='a'	0
{1,2,3,4,5,10} En el primer lugar	V="a" c='a'	1
{1,2,3,4,6,7,2,9,10} Cadena con solo un carácter, menor que el objetivo	V="a" c='b'	0
{1,2,3,4,6,8,2,9,10} Cadena con solo un carácter, mayor que el objetivo	V="b" c='a'	0



## Ejercicio 2

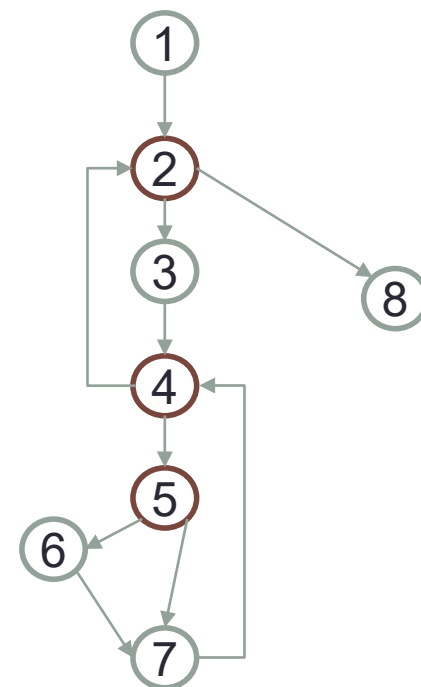
```
• static public void sort(int[] testArray)
• {
•     int tempValue;
•     int i = 0;
•     bool isSwapped = true;
•     while (isSwapped)
•     { isSwapped = false;
•       i++;
•       Console.Out.WriteLine("Before "+i+" iteration :");
•       Console.Out.WriteLine("");
•       for (int j = 0; j < testArray.Length - i; j++)
•       {
•           if (testArray[j] > testArray[j + 1])
•           {
•               tempValue = testArray[j];
•               testArray[j] = testArray[j + 1];
•               testArray[j + 1] = tempValue;
•               isSwapped = true;
•           }
•       }
•     }
• }
```

# Ejercicio 2

```

• static public void sort(int[] testArray)
• {
•   int tempValue;
•   ① int i = 0;
•   bool isSwapped = true;
•   while (isSwapped) ②
•   {   isSwapped = false;
•       i++;
•       ③ Console.WriteLine("Before "+i+" iteration :");
•       Console.WriteLine("");
•       for (int j = 0; j < testArray.Length - i; j++) ④
•       {
•           if (testArray[j] > testArray[j + 1]) ⑤
•           {
•               tempValue = testArray[j];
•               ⑥ testArray[j] = testArray[j + 1];
•               testArray[j + 1] = tempValue;
•               isSwapped = true;
•           }
•       }
•   }
•   ⑧

```



$V(G) = 4$

Áreas = 4

Nodos Predicado = 3  $\rightarrow 3 - 1 = 4$

Nodos = 8  $\rightarrow 10 - 8 + 2 = 4$

Aristas = 10

## Ejercicio 2

Path	Input	Output
{1,2,8} Cadena Vacía	No posible	No posible
{1,2,3,4,2,8} Vacía o una posición	[ ]	[ ]
{1,2,3,4,5,7,4,2,8} Dos posiciones ordenadas	[1,2]	[1,2]
{1,2,3,4,5,6,7,4,2,8} Dos posiciones desordenadas, pero siempre nos obliga a hacer una pasada más por el bucle, por lo que no puede probarse	No posible	No posible

