Búsqueda lineal
de un número en un arreglo ordenado

Objetivos. Resolver el problema de la búsqueda de un número en un arreglo ordenado de números usando el algoritmo de la búsqueda lineal. Practicar el ciclo `while` y prepararse a la idea de la búsqueda binaria que vamos a estudiar posteriormente.

Requisitos. Programación con el ciclo `while`, programación de funciones que trabajan con arreglos de números.

1. Los índices empiezan desde 1. En este texto se supone que los índices de los elementos de arreglos empiezan desde 1.

2. Ejemplo. Consideremos un arreglo a de números enteros y un número entero v:

$$a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7], \qquad v = 5.$$

Notamos que el arreglo está *ordenado de forma ascendente*:

$$a_1 \leq a_2 \leq \cdots \leq a_{10}.$$

Los índices de las componentes del arreglo a son $1, \ldots, 10$. Consideremos el conjunto de los índices j tales que $a_j \leq v$:

$$L = \{j \in \{1, \ldots, 10\} \colon\ a_j \leq v\} = \{1, 2, 3, 4, 5, 6, 7\}.$$

Escriba el número de elementos de L y el elemento máximo de L:

$$\#L = \underbrace{\qquad}_{?}, \qquad \max L = \underbrace{\qquad}_{?}.$$

3. Otro ejemplo. Consideremos el mismo arreglo a que en el ejemplo anterior, pero ahora ponemos $v = -2$:

$$a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7], \qquad v = -2.$$

Encontramos el conjunto de los índices j tales que $a_j \leq v$:

$$L = \{j \in \{1, \ldots, 10\} \colon\ a_j \leq v\} = \underbrace{\qquad\qquad\qquad\qquad}_{?}.$$

Escriba el número de elementos de L y el elemento máximo de L:

$$\#L = \underbrace{\qquad}_{?}, \qquad \max L = \underbrace{\qquad}_{?}.$$

4. Relación entre la cantidad de los índices y el índice máximo. Sea a un arreglo de números enteros ordenado de manera ascendente y sea v un número entero. Denotemos por n a la longitud del arreglo a y por L al conjunto de los índices j tales que $a_j \le v$:

$$L = \{j \in \{1, \ldots, n\}: \ a_j \le v\}.$$

Analice cómo está relacionado el tamaño del conjunto L con el valor del máximo elemento del conjunto L.

$$\max(L) \underbrace{\qquad}_{?} \#L.$$

Notemos que la cantidad $\#L$ está relacionada con el concepto de la *función de distribución* en la teoría de probabilidad.

5. Idea de la búsqueda lineal. Ir desde el inicio del arreglo a, comparando sus entradas con el número dado v. Organizarlo como un ciclo de tipo `while`. Denotar por j al índice del elemento del arreglo. Aumentar j en cada paso. Hay que terminar cuando el índice actual ya está fuera del conjunto $\{1, \ldots, n\}$ de índices admitibles o cuando a_j es estrictamente mayor que v:

$$\text{Condición de terminación:} \qquad (j \ge n) \qquad \vee \qquad (a_j > v).$$

Escriba bien la **condición de continuación**:

Condición de continuación:

6. Sobre la operación lógica "y". En la lógica matemática, la operación lógica "y" es conmutativa, pero en programación el cálculo de las expresiones `p && q` y `q && p` se puede realizar de manera diferente. En algún lenguaje de programación escriba la siguiente función f. La función no es pura, es decir, además de trabajar con variables locales y regresar un valor, la función cambia el mundo global de alguna manera. A saber, nuestra función f siempre produce un mensaje en la pantalla:

```
fun f(a):
   print('cogito ergo sum');
   b <- (remainder(a, 5) == 0);
   return b;
```

Haga las siguientes pruebas de esta función:

```
f(3)
f(5)
True && f(11)
f(11) && True
False && f(11)
f(11) && False
```

7. El orden de las condiciones en la condición compuesta del ciclo while. Supongamos que a es un arreglo de longitud 10 y la variable j tiene valor 11. Determine cuál de las siguientes dos expresiones va a generar un error:

```
(a[j] <= 7) && (j <= 10)
```

```
(j <= 10) && (a[j] <= 7)
```

Regresamos el Ejercicio 5. Piense cómo escribir la condición compuesta del ciclo `while`.

8. Problema. En algún lenguaje de programación escribir una función `linearsearch` que determine el número de las componentes de un arreglo ordenado que son menores o iguales a un número dado.

Entrada: un arreglo a de números enteros (denotemos la longitud de este arreglo por n) y un número entero v.

Propiedades de la entrada: se supone que las componentes del arreglo a están ordenadas de manera ascendente (en el sentido no estricto):

$$a_1 \leq a_2 \leq \ldots \leq a_n.$$

Salida: la cantidad de los elementos del conjunto

$$\{j \in \mathbb{Z}\colon\ 1 \leq j \leq n,\ a_j \leq v\}.$$

Esquema:

```
fun linearsearch(a):
   n <- length(a);
   j <- ???;
   while (???)
      j <= j + 1;
   return j
```

9. Pruebas. Haga varias pruebas de la función programada, incluso pruebas con los siguientes datos:

- $a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7]$, $v = 5$.
- $a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7]$, $v = -2$.
- $a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7]$, $v = -7$.
- $a = [-3, -2, -2, 0, 4, 4, 4, 6, 7, 7]$, $v = 9$.
- $a = [\,]$, $v = 5$. En este caso el arreglo a es vacío, $n = 0$.