Ejercicio 3

Diseñe una variación de Búsqueda Binaria (algoritmo 1.4) que efectúe sólo una comparación binaria (es decir, la comparación devuelve un resultado booleano) de K con un elemento del arreglo cada vez que se invoca la función. Pueden hacerse comparaciones adicionales con variables de intervalo. Analice la corrección de su procedimiento. Sugerencia: ¿Cuándo deberá ser de igualdad (==) la única comparación que se hace?

Análisis

Extracción de Datos y Condiciones Iniciales

- **Entrada:** Un conjunto ordenado $S = \{a_0, a_1, \ldots, a_{n-1}\}$ donde se cumple que $a_i \leq a_{i+1}$ para todo índice válido.

- **Objetivo (K):** Valor entero a buscar dentro de S.

- **Restricción del problema:** Se prohíbe el uso de la estructura clásica $if\ (x < K) \ldots else\ if\ (x > K) \ldots else\ return$, ya que esto implica dos comparaciones por ciclo.

Lógica de Partición (El Invariante del Bucle)

A diferencia de la búsqueda binaria tradicional que divide el espacio en tres ($<, >, =$), este algoritmo particiona el espacio de búsqueda en **dos subconjuntos** complementarios en cada paso:

1. **Conjunto A:** Elementos estrictamente menores que K $(x < K)$.
2. **Conjunto B:** Elementos mayores o iguales a K (x ≥ K).

El Invariante:

La propiedad que se mantiene verdadera durante toda la ejecución del bucle es:

"Si el elemento K existe en el arreglo original, entonces su índice se encuentra necesariamente dentro del intervalo actual [L, R]."

Análisis de las Operaciones y Convergencia

Analizamos el comportamiento matemático dentro del bucle $while\ (izquierda < derecha)$:

1. **Cálculo del Pivote (m):**

$$m = L + \left\lfloor \frac{R - L}{2} \right\rfloor$$

La función piso $\lfloor\cdot\rfloor$ asegura que m siempre tiende hacia la izquierda. Esto es crucial para evitar bucles infinitos cuando L y R son adyacentes.

2. **La Comparación Única** ($a_m < K$):

   Se evalúa si el elemento central pertenece al "Conjunto A" (menores estrictos).

   - **Caso Verdadero** ($\boldsymbol{a_m < K}$):

     Matemáticamente, si $a_m < K$, entonces por la propiedad de orden, todo elemento a_i con $i \leq m$ es también menor que K. Por lo tanto, K no puede estar en el índice m ni a su izquierda.

     - **Acción:** Reducimos el intervalo por la izquierda:
       $$L_{nuevo} = m + 1.$$
   - **Caso Falso** ($\boldsymbol{a_m \geq K}$):

     Esto implica que a_m es mayor o igual a K. No sabemos si es igual, pero sabemos con certeza que K no puede estar a la derecha de m (dado que el arreglo es creciente, cualquier índice $j > m$ tendría $a_j \geq a_m \geq K$)

     - **Acción:** Reducimos el intervalo por la derecha, pero conservando m como posible candidato: $R_{nuevo} = m$.
     -

3. **Terminación del Bucle:**

   El bucle termina estrictamente cuando $L = R$. En este punto, el intervalo se ha reducido a un tamaño de 1. No quedan más elementos por descartar.

Respuesta a la Sugerencia del Problema

El enunciado pregunta: *¿Cuándo deberá ser de igualdad (==) la única comparación que se hace?*

Respuesta: La comparación de igualdad se realiza **única y exclusivamente al salir del bucle**. Dado que el algoritmo garantiza que, si K existe, estará "atrapado" en la posición $L = R$, solo necesitamos una verificación final: $arr[L] == K$. Si esta comparación es verdadera, hemos encontrado el elemento; de lo contrario, el elemento no existe.