

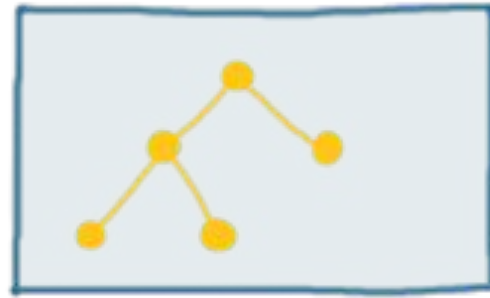
Búsqueda Binaria

Esta presentación proporciona una exploración en el concepto de búsqueda binaria, su complemento con el ordenamiento. Cubre ejemplos y técnicas.

Aclaración: Tipos de estructuras



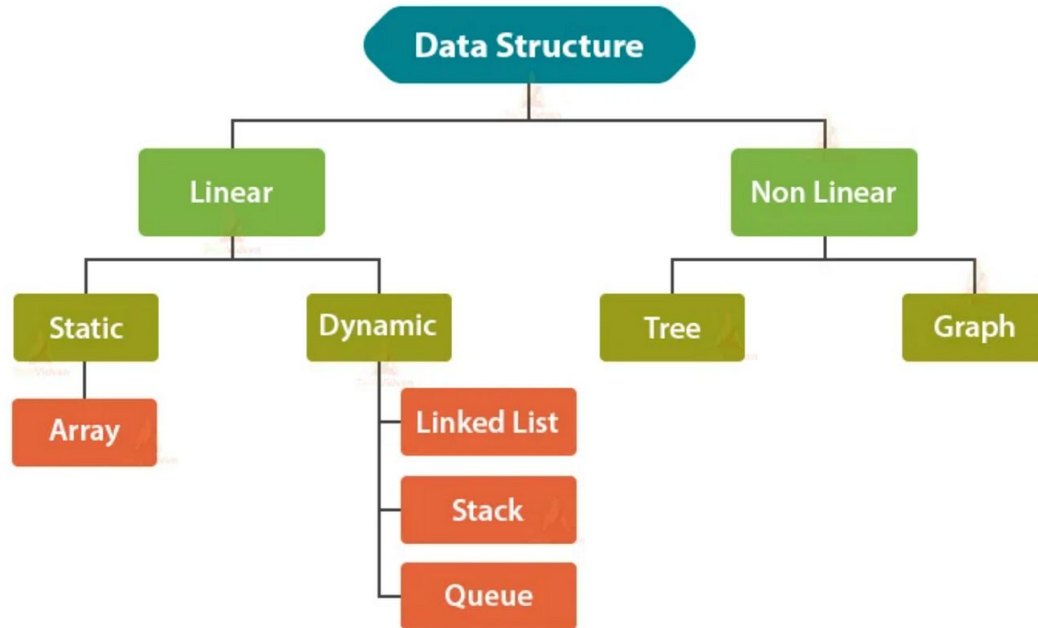
Linear



Non-Linear

Aclaración: Tipos de estructuras

Data Structure Classification in Java

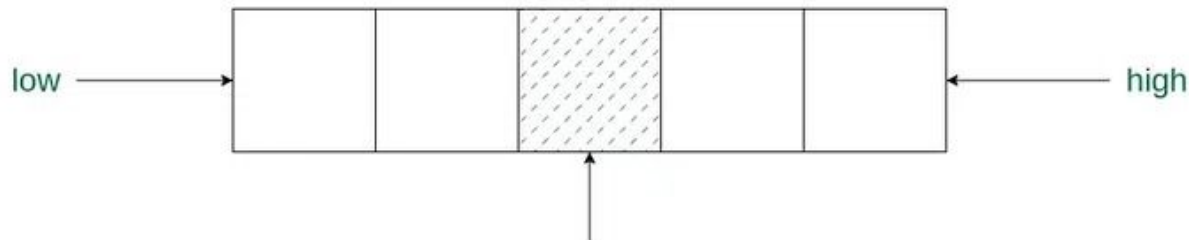


¿Para qué ordenamos elementos?

- El proceso de búsqueda de un elemento en una estructura depende básicamente de si éste se encuentra ordenado. En caso de que no haya ningún orden en los valores, la única opción que tenemos es hacer una búsqueda secuencial utilizando para esto el patrón de recorrido parcial clásico.
- El ordenamiento facilita la búsqueda de elementos pues permite aplicar técnicas para reducir la complejidad temporal de buscar un elemento.

¿Para que los usamos?

Si la estructura en la que queremos buscar un elemento se encuentra ordenada, podemos utilizar una técnica muy eficiente de localización que se denomina la búsqueda binaria.



$$\text{mid} = \text{low} + (\text{high} - \text{low})/2$$

Búsqueda Binaria: Algoritmo

1. Ordenar el arreglo
2. Escoger los valores para Low y High
3. Escoger el valor de Mid
4. Buscar con la técnica de divide y vencerás.

Search for 47

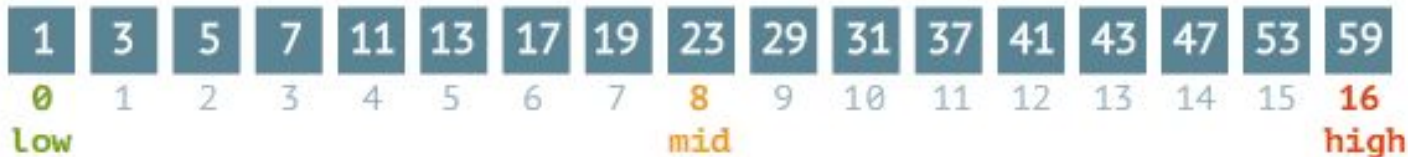
0	4	7	10	14	23	45	47	53
---	---	---	----	----	----	----	----	----

```
function binarysearch(list,element):  
    left = 0  
    right = length(list) - 1  
    while left <= right:  
        mid = (left + right) // 2  
        if list[mid] == target:  
            return mid  
        elif list[mid] < target:  
            left = mid + 1  
        else:  
            right = mid - 1  
    return -1
```

Comparación: Búsqueda Lineal (secuencial) vs Búsqueda Binaria

Binary search

steps: 0



Sequential search

steps: 0



Comparación: Complejidad Temporal de algoritmos populares de búsqueda

TIME COMPLEXITY

ALGORITHM	BEST CASE	AVERAGE CASE	WORST CASE
LINEAR SEARCH	$O(1)$	$O(N)$	$O(N)$
BINARY SEARCH	$O(1)$	$O(\log_2 N)$	$O(\log_2 N)$
TERNARY SEARCH	$O(1)$	$O(\log_3 N)$	$O(\log_3 N)$
JUMP SEARCH	$O(1)$	$O(\sqrt{N})$	$O(\sqrt{N})$
INTERPOLATION SEARCH	$O(1)$	$O(\log(\log(N)))$	$O(N)$

Tarea: Implementar el algoritmo de búsqueda binaria y buscar un elemento Computador dentro de un arreglo ordenado, teniendo en cuenta la hoja de trabajo de ordenamiento.

Gracias por su tiempo y atención 😊