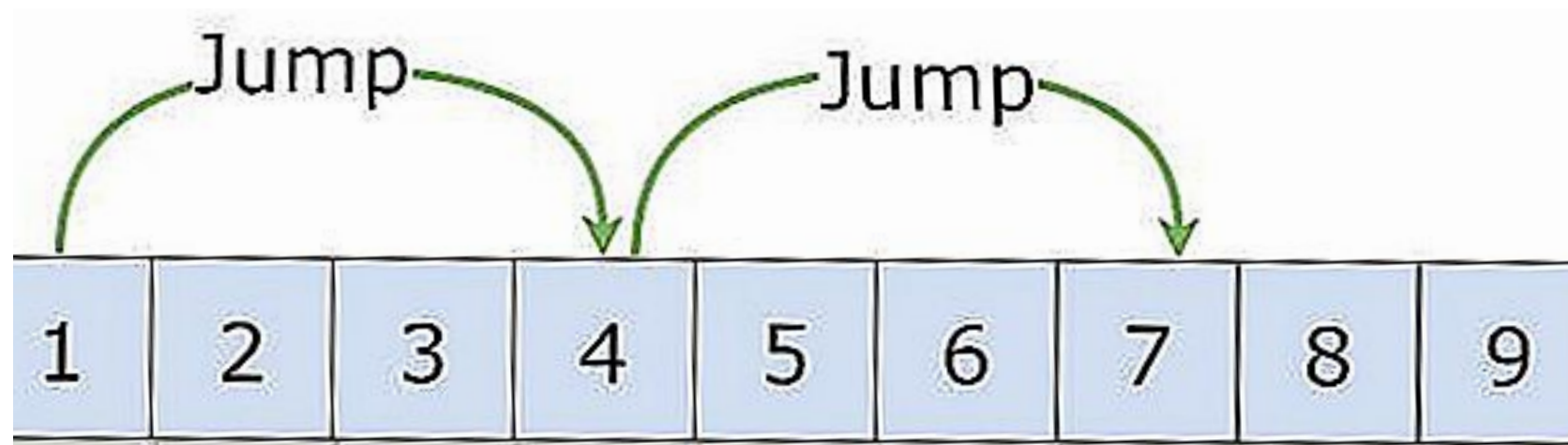


MÁS ALGORITMOS DE BÚSQUEDA

```
    }  
    render() {  
      return (  
        <React.Fragment>  
          <div className="py-5">  
            <div className="container">  
              <Title name="our" title="product">  
                <div className="row">  
                  <ProductConsumer>  
                    {(value) => {  
                      console.log(value)  
                    }}  
                  </ProductConsumer>  
                </div>  
              </div>  
            </div>  
          </React.Fragment>  
        )  
      )  
    }  
  }  
}
```


Es un algoritmo de búsqueda que utiliza el salto en lugar de realizar comparaciones lineales para buscar un elemento en una lista ordenada.

No se enfoca en buscar elemento por elemento si no en ahorrar tiempo al buscar dentro del arreglo.



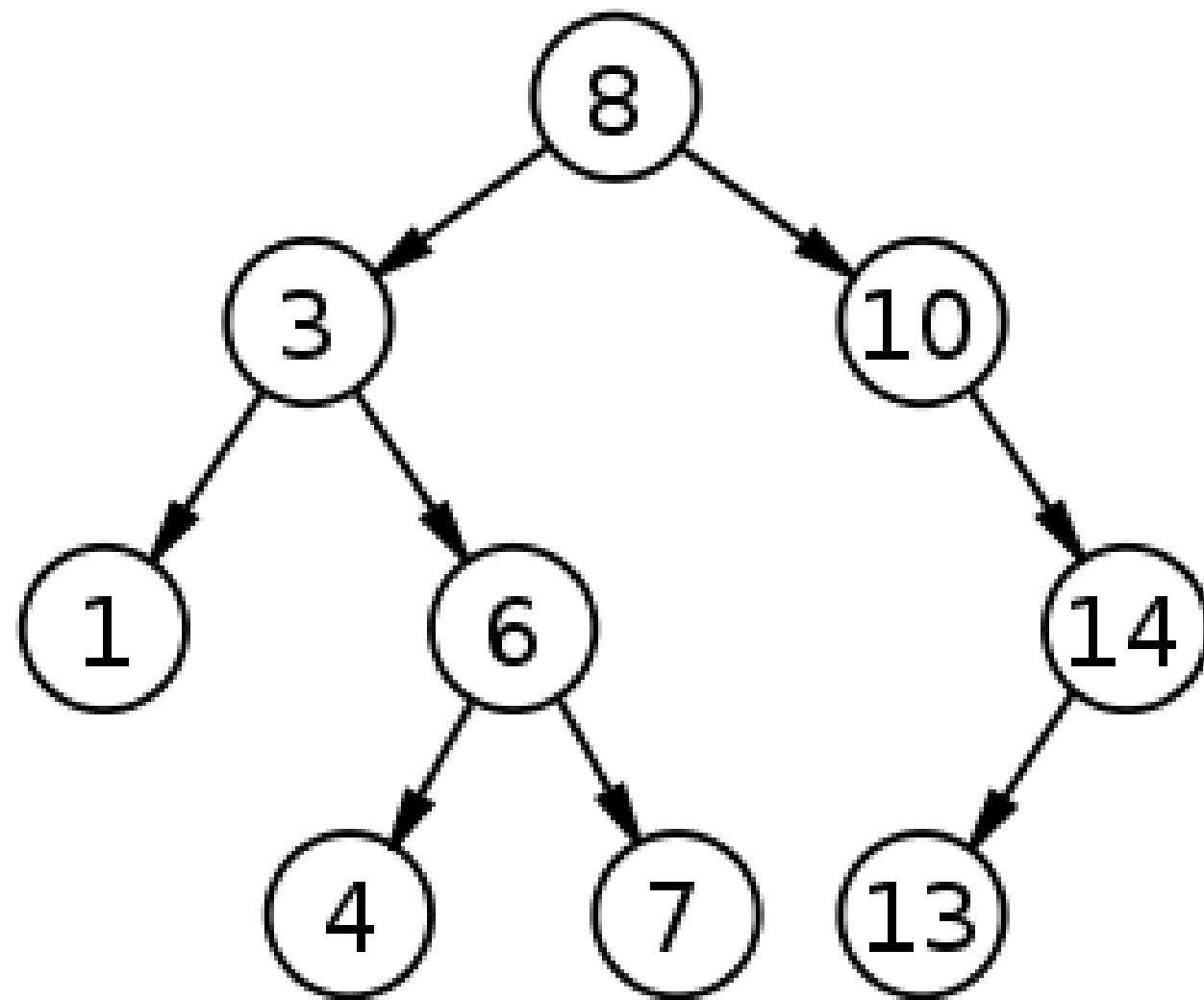
BÚSQUEDA POR SALTO


```
// Función para realizar la búsqueda por salto
int salto(int arr[], int n, int busqueda ) {
    int salto = sqrt(n);
    int prev = 0;
    // Buscar el bloque que contiene el elemento buscado
    while (arr[min(salto, n) - 1] < busqueda ) {
        prev = salto;
        salto += sqrt(n);
        if (prev >= n) {
            // Si el bloque actual es mayor o igual que el tamaño del arreglo,
            // significa que el elemento no está presente en la lista.
            return -1;
        }
    }
    // Realizar una búsqueda lineal dentro del bloque actual
    for (int i = prev; i < min(salto, n); ++i) {
        if (arr[i] == busqueda) {
            return i; // Elemento encontrado, devolver el índice
        }
    }
    return -1; // Elemento no encontrado
}
```

BÚSQUEDA POR SALTO



Se refiere a una técnica de búsqueda y organización de datos utilizando estructuras de árboles binarios. Un árbol binario es una estructura de datos en la que cada nodo tiene hasta dos hijos: un hijo izquierdo y un hijo derecho.



BÚSQUEDA POR ÁRBOLES BINARIOS

Para la creación inicial del árbol binario, se debe comenzar por la definición de una estructura:

```
struct nodo{  
    int dato;  
    struct nodo *der;  
    struct nodo *izq;  
};
```

Los siguientes pasos serán datos en código.

BÚSQUEDA POR ÁRBOLES BINARIOS



EJERCICIO

Crear una estructura con los datos para un vehículo.

Hacer un menú con las siguientes opciones:

1. Crear un vehículo con todos sus datos. Mostrar algunos como "placa", "dueño", "vin" etc.
2. Crear una cantidad "n" de vehículos (donde n es un número entero ingresado por el usuario). Crear las siguientes opciones para este submenú:
 - a. Mostrar los datos en el orden en el que fueron ingresados
 - b. Mostrar los datos al revés de como fueron ingresados
 - c. Realizar la búsqueda por placa de un vehículo.

EJERCICIOS

