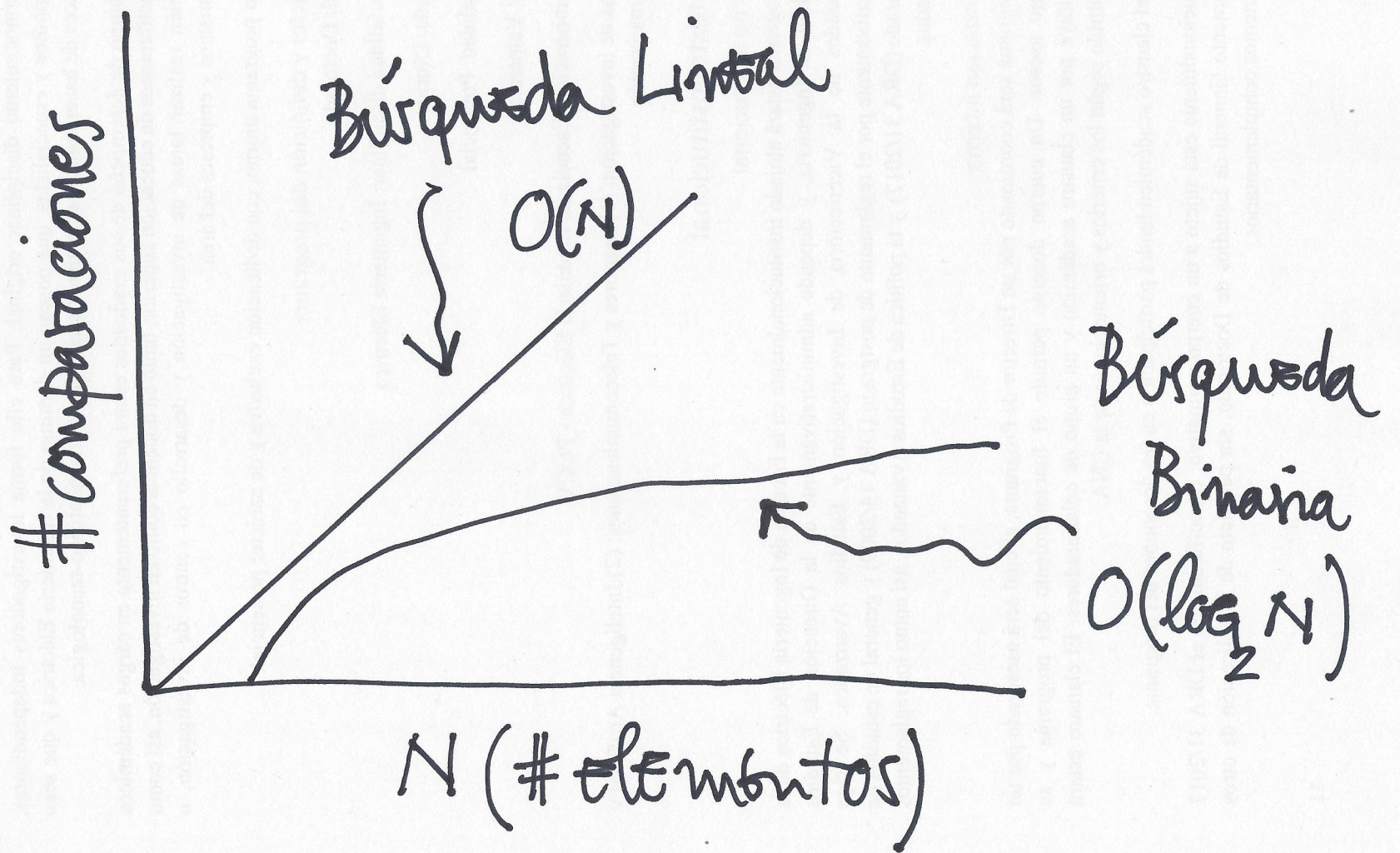
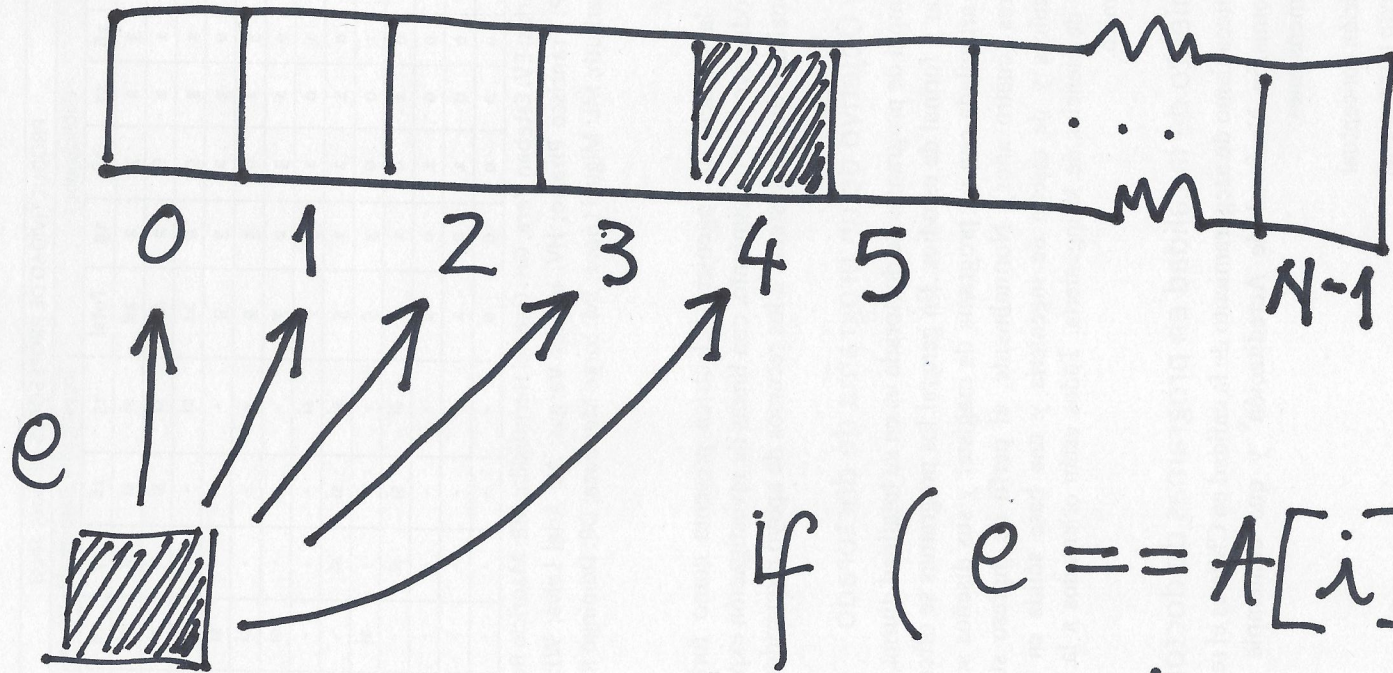


Búsqueda en ARREGLOS



Búsqueda Lineal

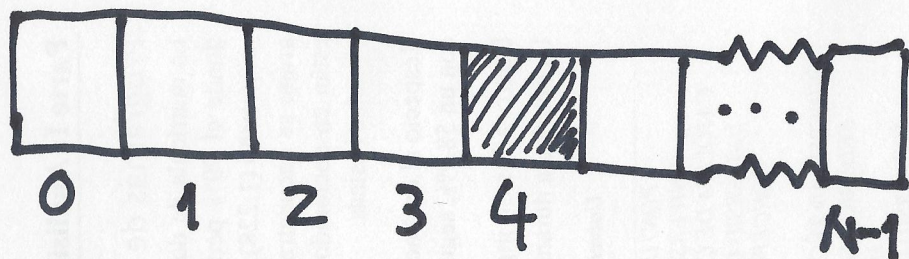
A



```
if (e == A[i])  
    ok!  
else  
    i++;
```

Búsqueda Lineal

A



e



```
int i, flag;  
i = 0; flag = 0;  
while (i < N) && (!flag) {  
    if (e == A[i])  
        flag = 1;  
    else  
        i++;  
}  
if (flag)  
    printf("Encontrado!\n");  
else  
    printf("No encontrado!\n");
```

Búsqueda Lineal

```
int BLineal (int *, int, int);
```

```
int main() {
```

```
    int A[6] = {2, 4, 6, 8, 10, 12};
```

```
    int e = 41;
```

```
    if (BLineal(A, 6, e) == 1) ←
```

```
        printf("encontrado! \n");
```

```
    ELSE
```

```
        printf("NO encontrado! \n");
```

```
    return 0;
```

```
}
```

```
int Blineal (int *A, int n  
            , int e) {
```

```
    ...
```

```
    if (flag)
```

```
        RETURN 1;
```

```
    ELSE
```

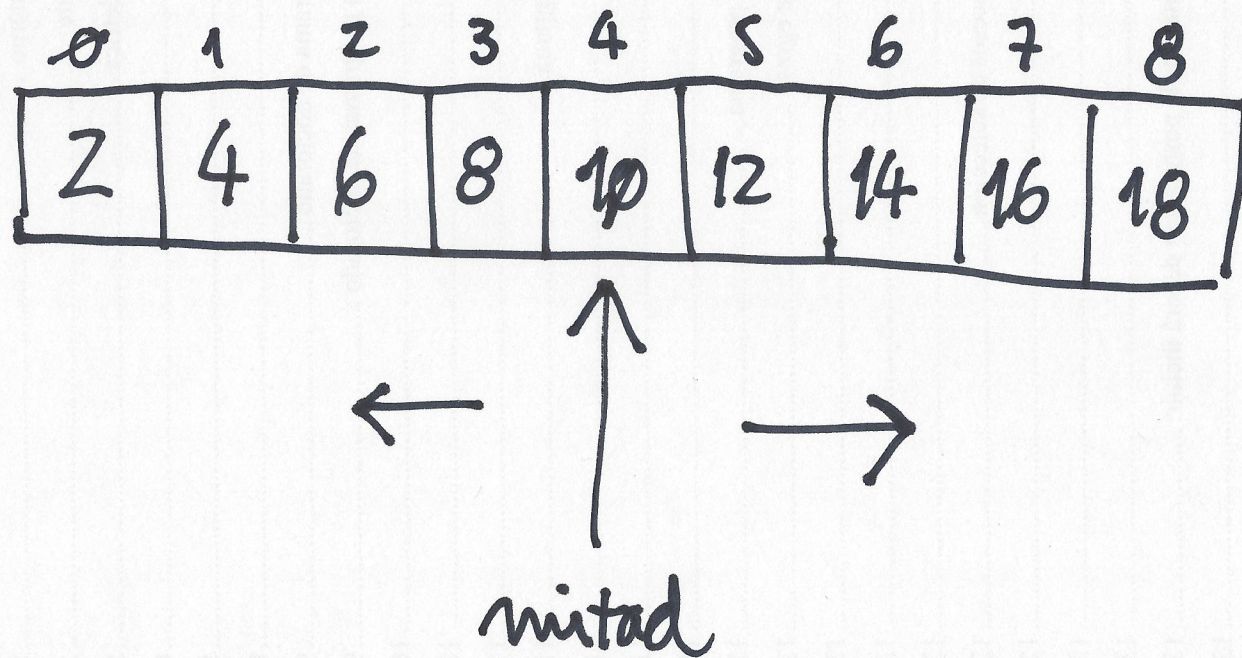
```
        return 0;
```

```
}
```

Búsqueda Binaria

* Es más eficiente.

* Los datos deben estar ordenados.

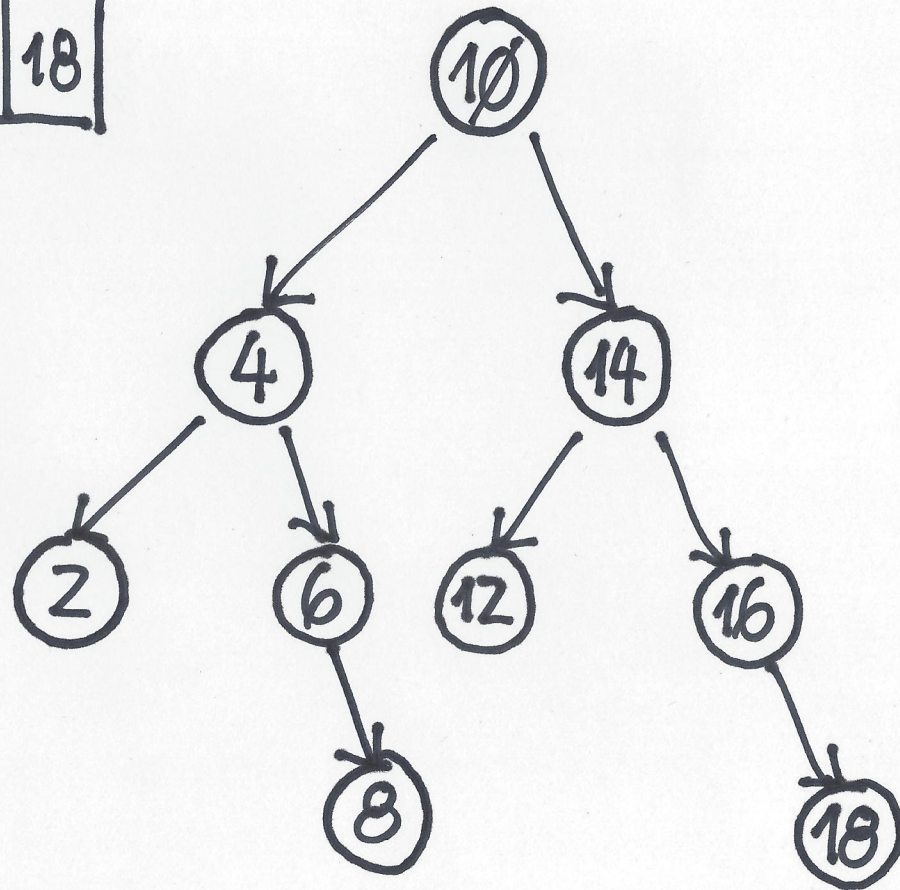


Búsqueda Binaria

0	1	2	3	4	5	6	7	8
2	4	6	8	10	12	14	16	18



mitad



#comparaciones
 $\log_2(9)$

Busqueda Binaria

A

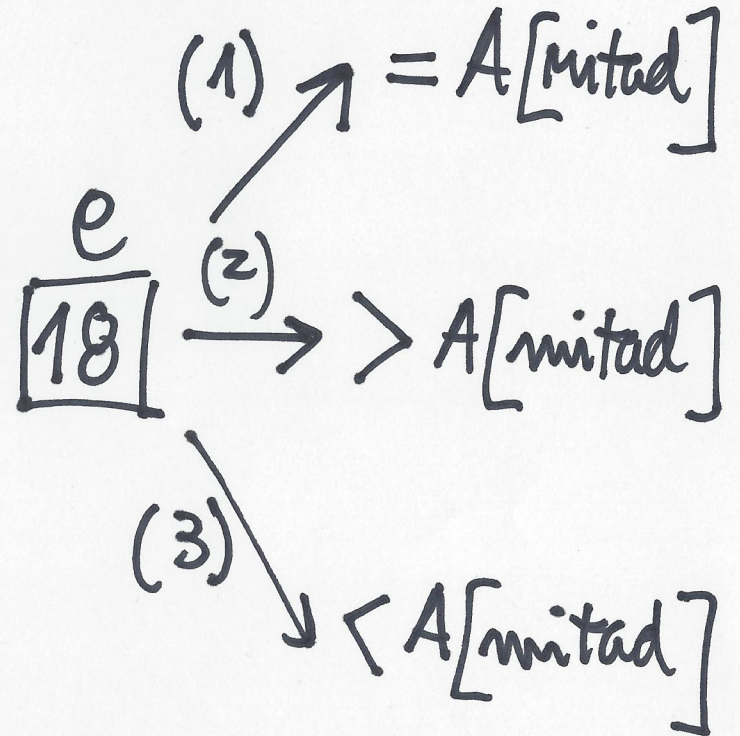
0	1	2	3	4	5	6	7	8
2	4	6	8	10	12	14	16	18

↑
$$\text{mitad} = \frac{(8+0)}{2}$$

A

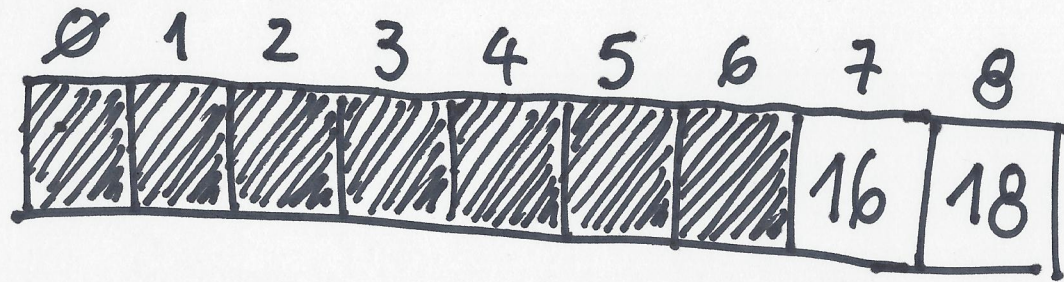
0	1	2	3	4	5	6	7	8
					12	14	16	18

↑
$$\text{mitad} = \frac{(8+5)}{2}$$



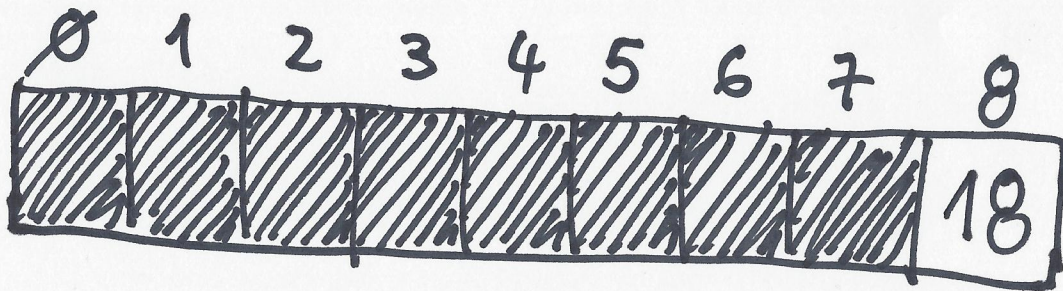
Búsqueda Binaria

A



$$\text{mitad} = \left(\frac{7+8}{2} \right)$$

A



Encontrado! ▽