

Data Structures

# Searching Technique – Binary Search



CODE  
FOR THINGS

# Algorithm - Binary Search Iterative

# Data Structure – Binary Search Iterative

# Algorithm

## **Binary\_search\_iterative(arr, key)**

### **Input Specification:**

arr : Array to hold the elements  
key : Item to be searched

### **Output Specification:**

mid : Positon of key element  
-1 : Key is not found



# Binary\_search\_iterative(arr,key)

arr[SIZE]

SIZE = 10



# Binary\_search\_iterative(arr,key)

arr[SIZE]

SIZE = 10

key=12

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]

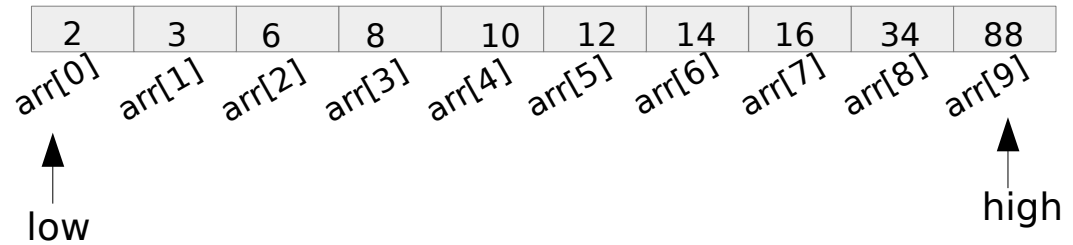
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



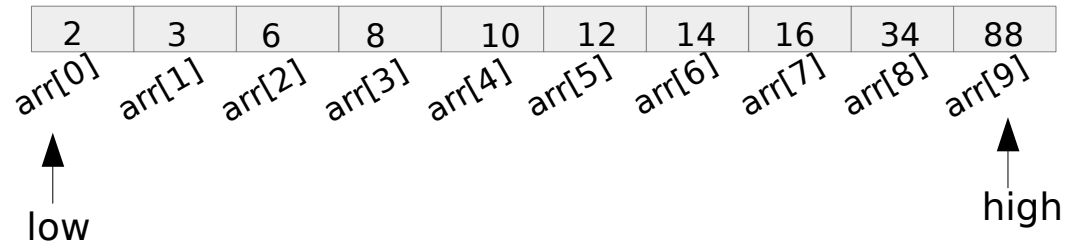
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12





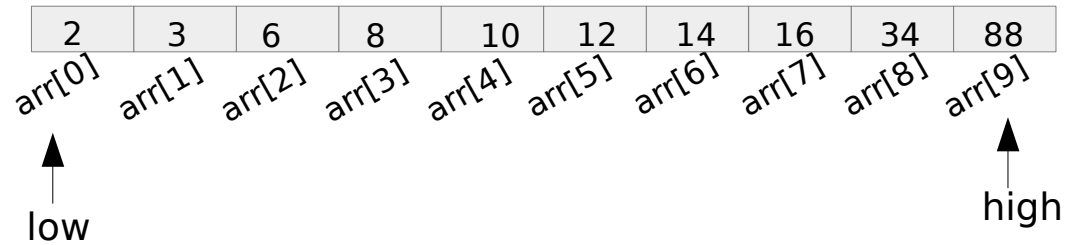
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



mid

$$(0+9)/2 = 4$$

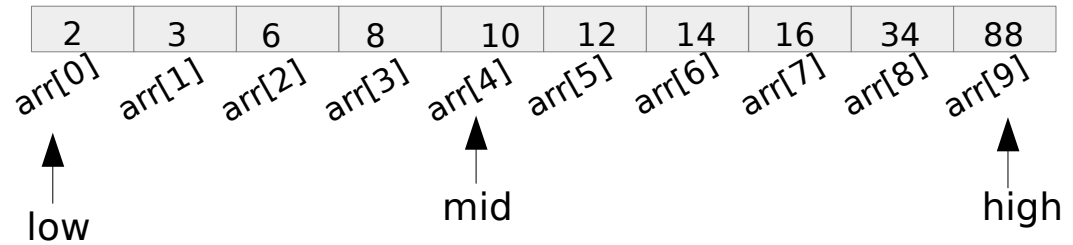
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



mid

$$(0+9)/2 = 4$$

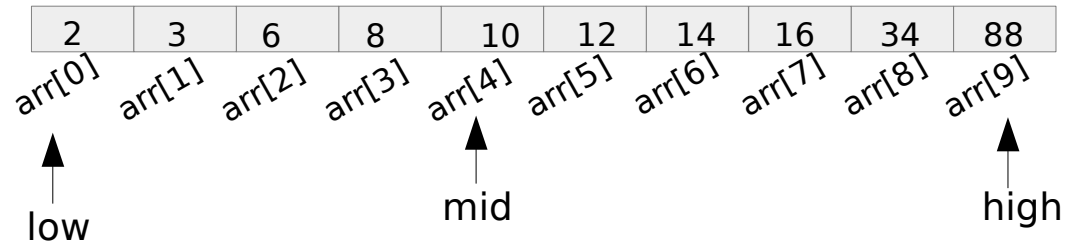
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12

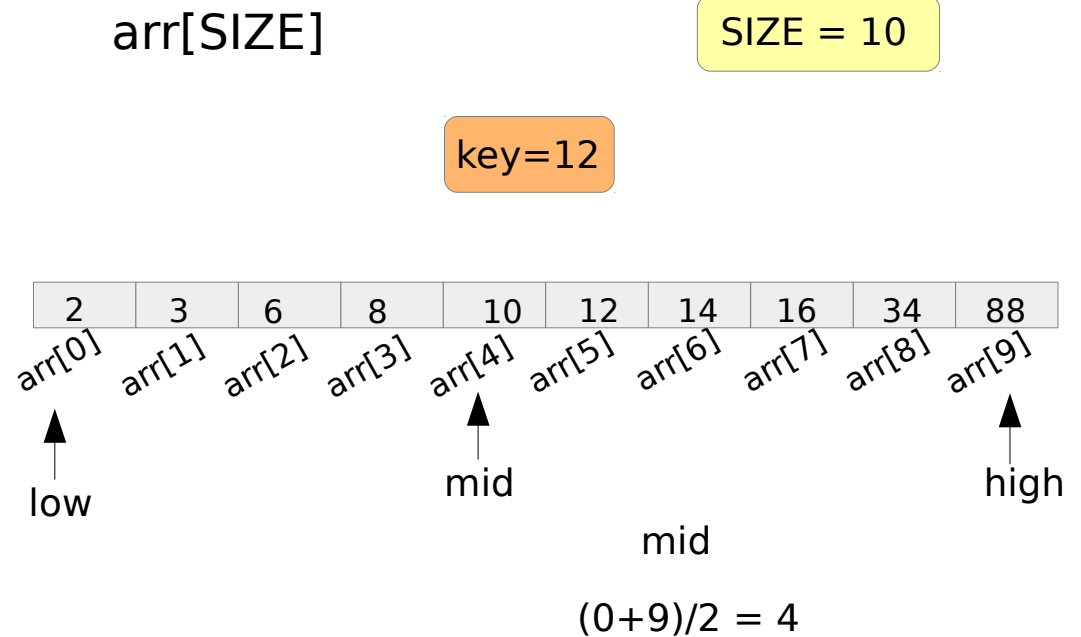


mid

$$(0+9)/2 = 4$$

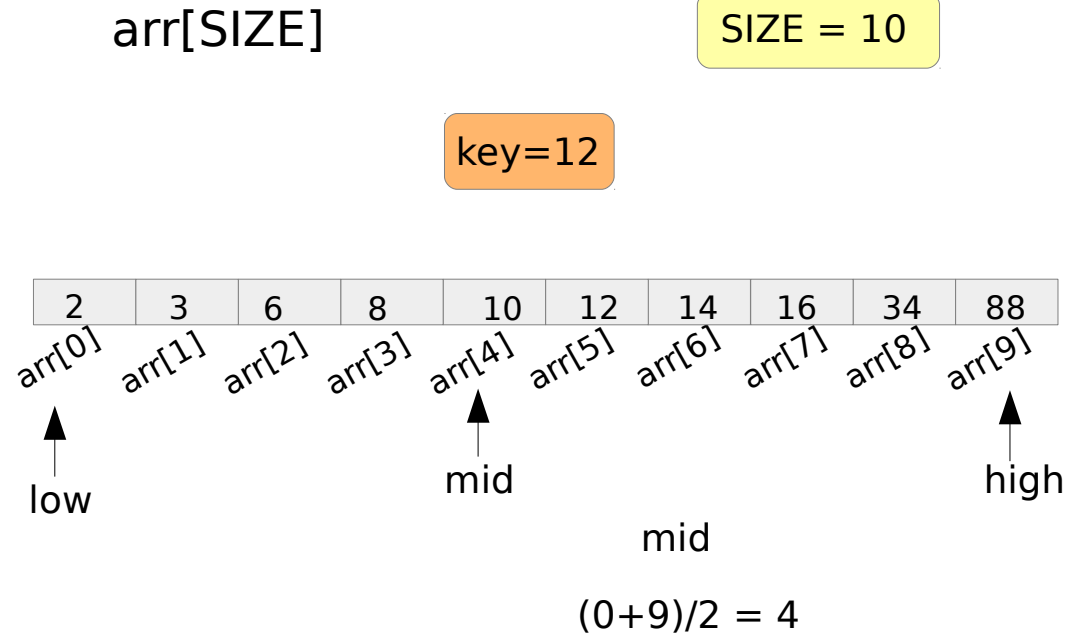
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```



# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```



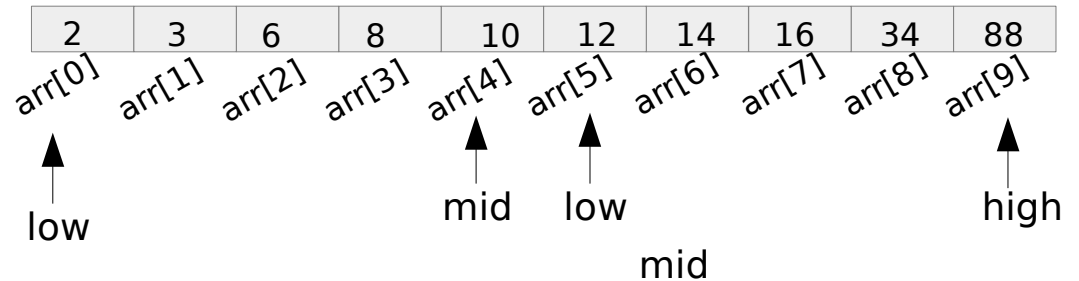
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

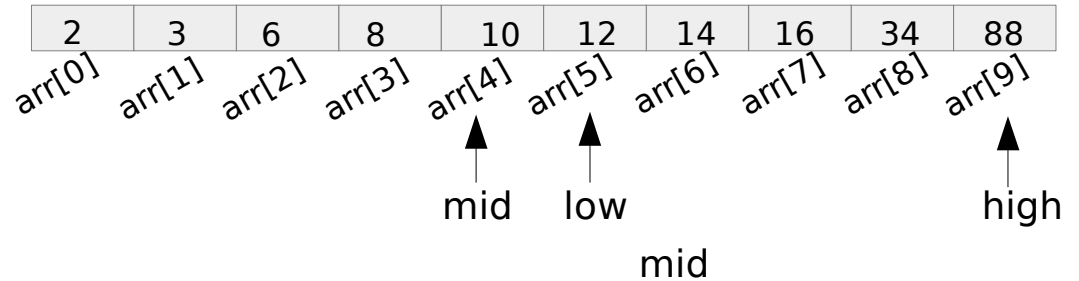
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

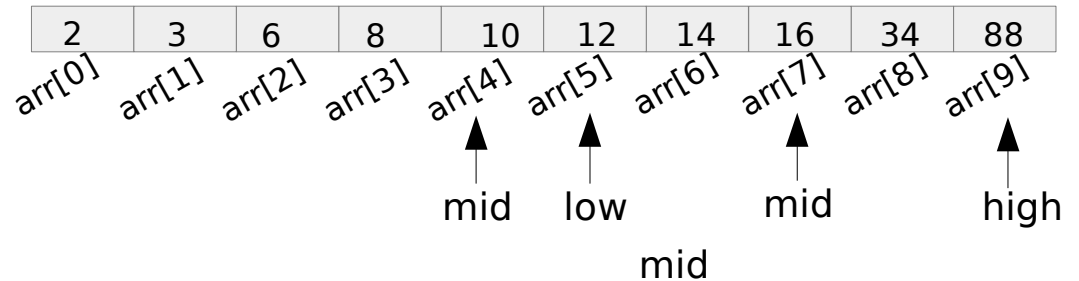
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$



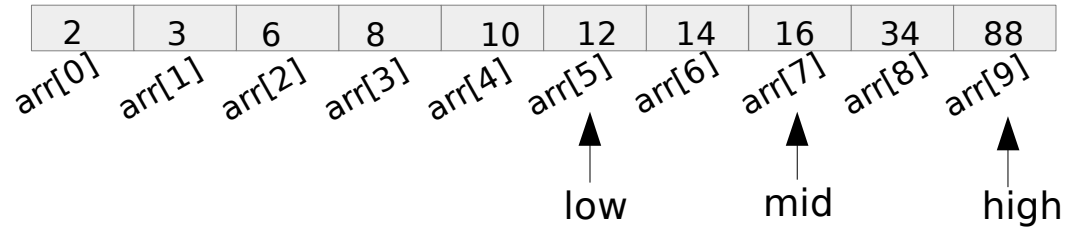
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



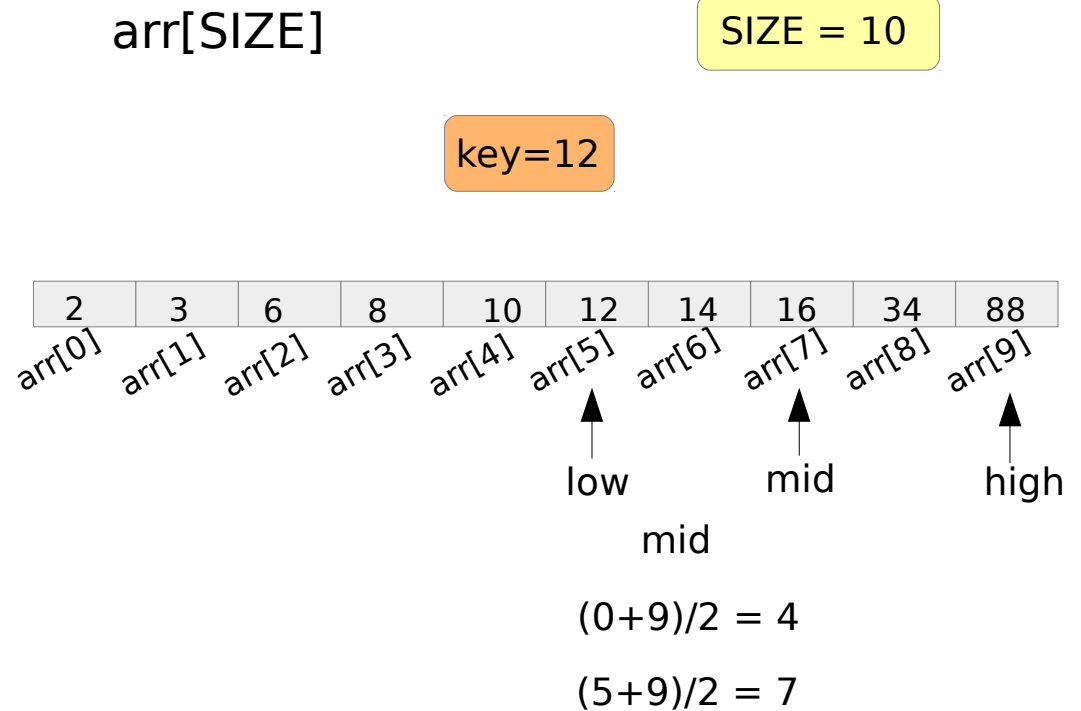
mid

$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```



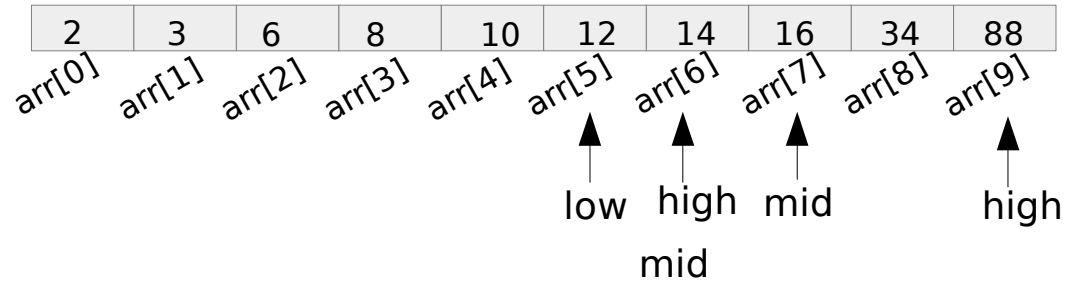
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

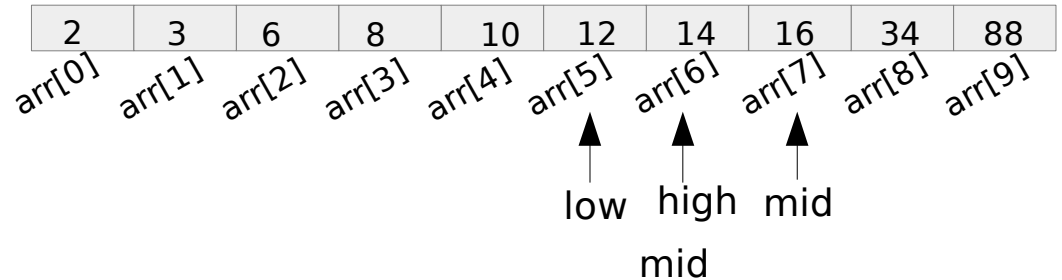
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

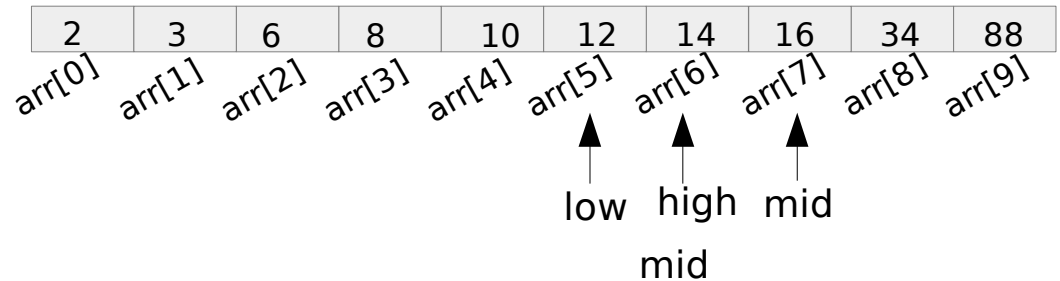
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

$$(5+6)/2 = 5$$

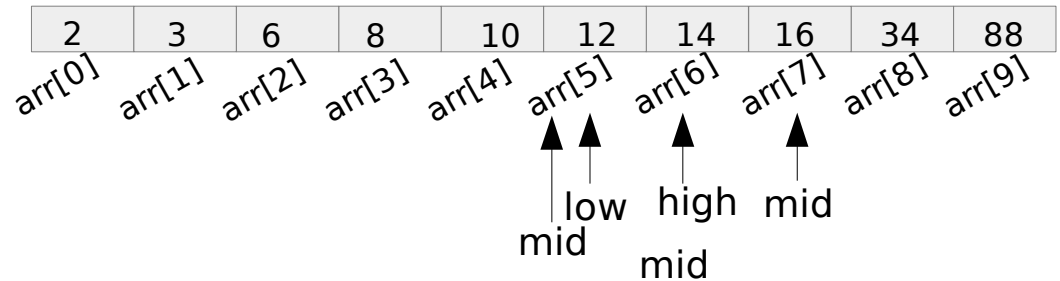
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

$$(5+6)/2 = 5$$

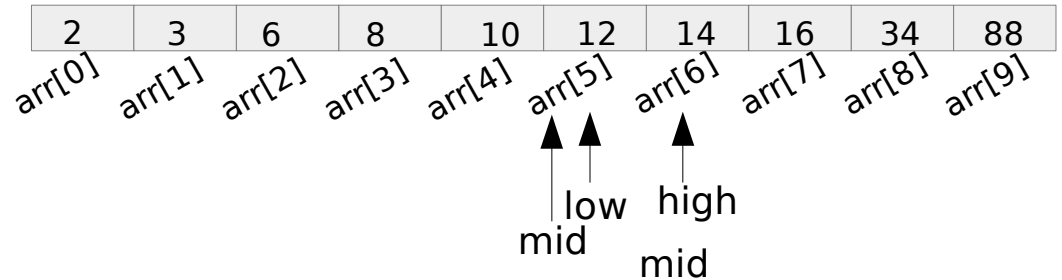
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

$$(5+6)/2 = 5$$

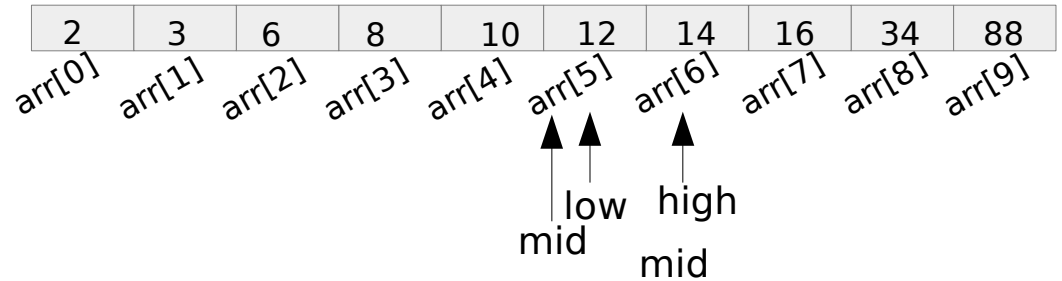
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

$$(5+6)/2 = 5$$



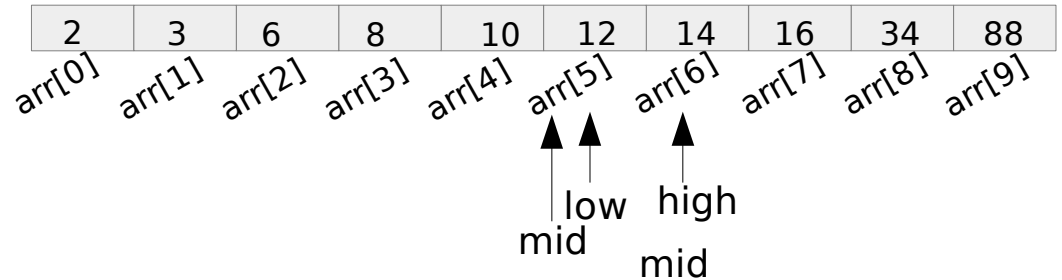
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=12



mid=5

$$(0+9)/2 = 4$$

$$(5+9)/2 = 7$$

$$(5+6)/2 = 5$$

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]

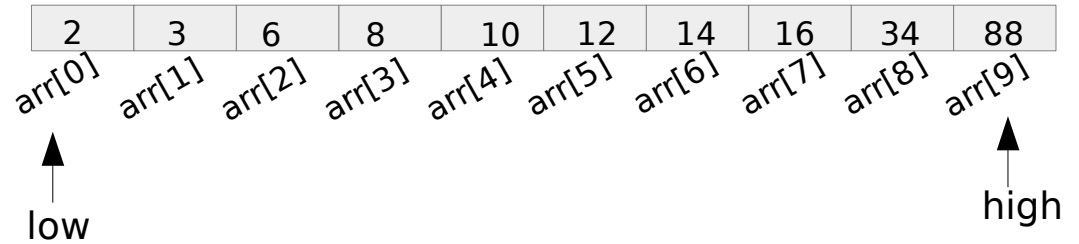
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



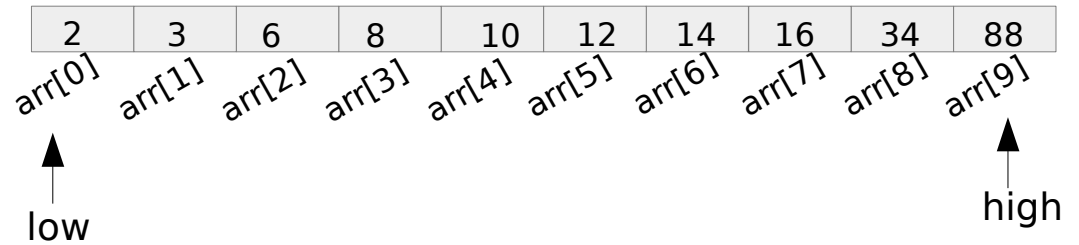
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



```
low = 0 , high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

SIZE = 10

2	3	6	8	10	12	14	16	34	88
---	---	---	---	----	----	----	----	----	----

arr[0] arr[1] arr[2] arr[3] arr[4] arr[5] arr[6] arr[7] arr[8] arr[9]

low high

$$(0+9)/2 = 4$$

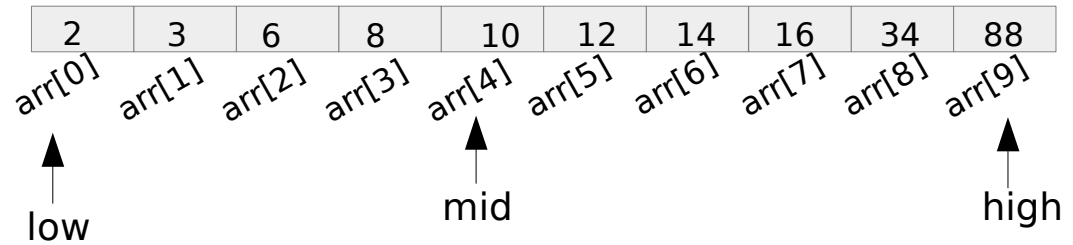
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



mid

$$(0+9)/2 = 4$$

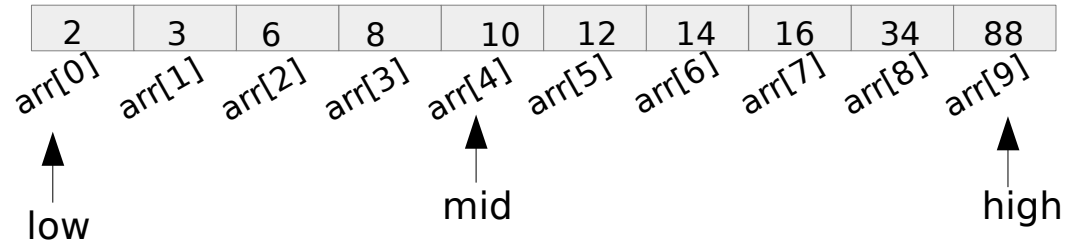
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

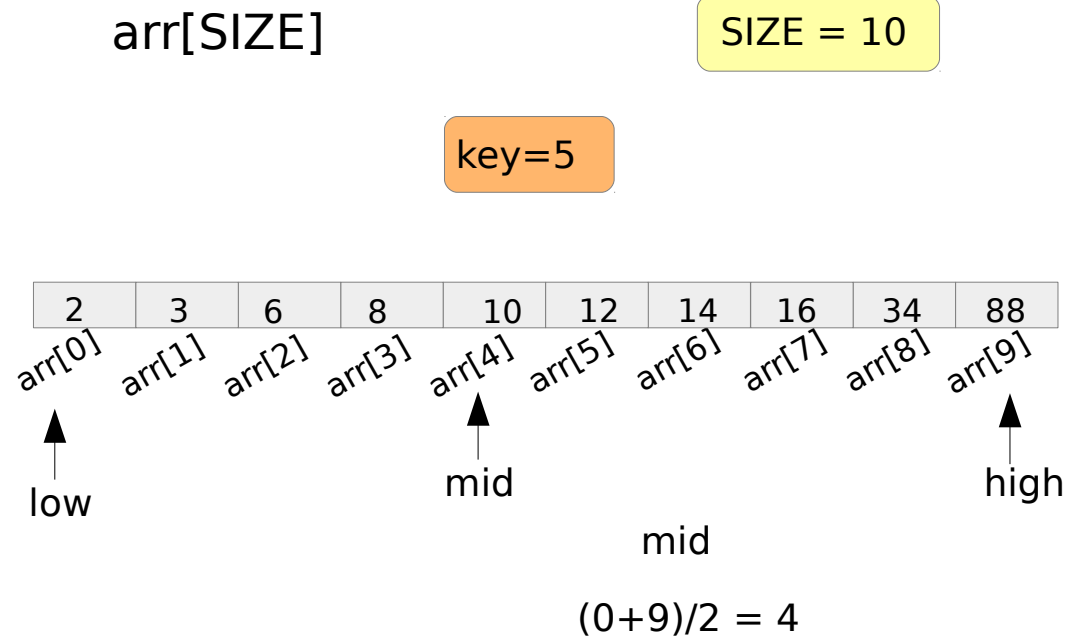


mid

$$(0+9)/2 = 4$$

# Binary\_search\_iterative(arr,key)

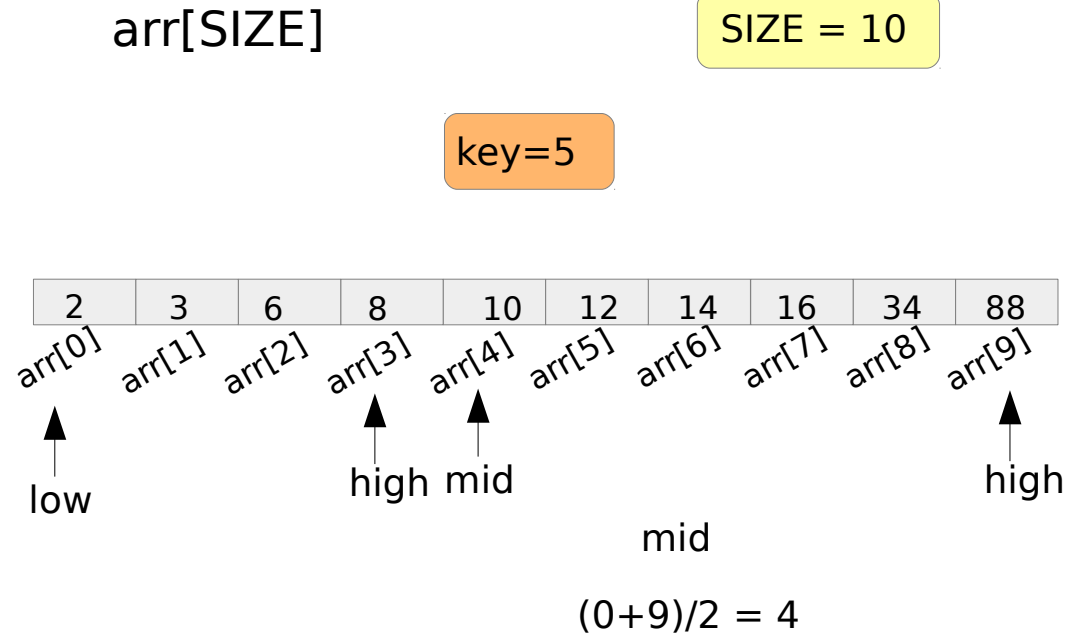
```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```





# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```



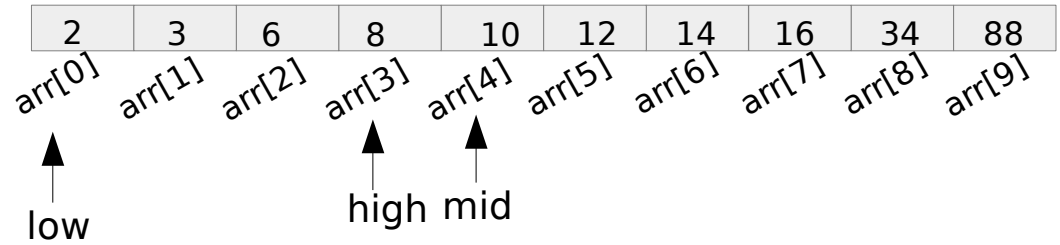
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



mid

$$(0+9)/2 = 4$$

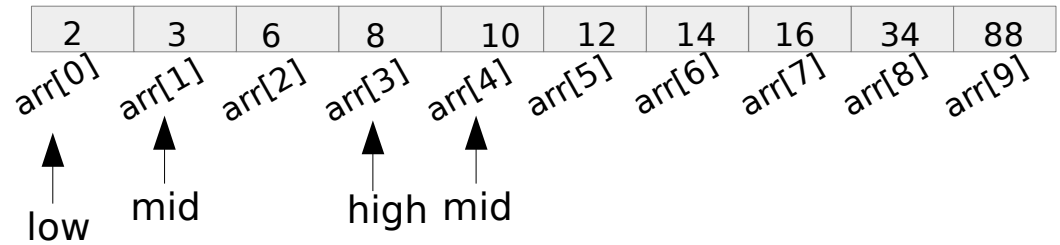
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

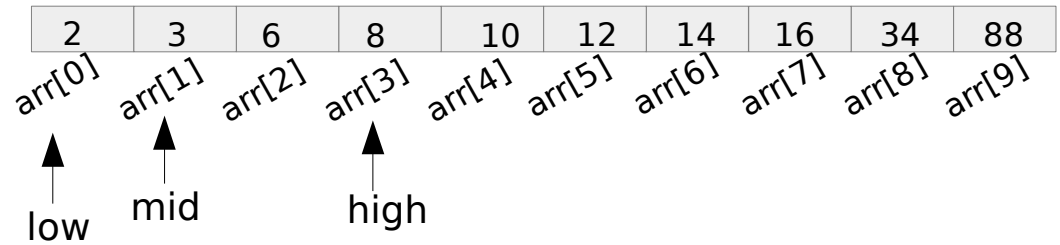
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

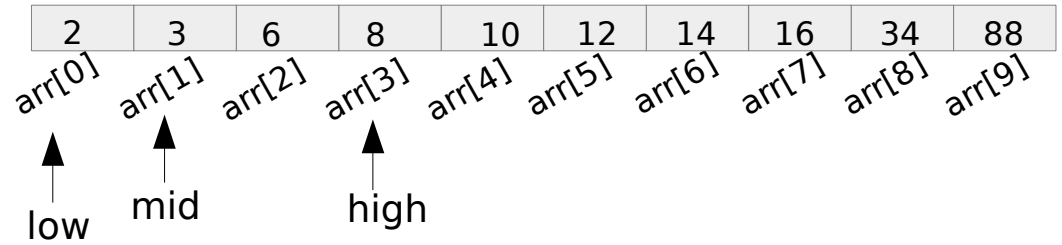
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
↑	↑	↑							
low	mid	high							

mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
↑	↑	↑	↑						
low	mid	low	high						

mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
	↑	↑	↑						
	mid	low	high						

mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$



# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5

2	3	6	8	10	12	14	16	34	88
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
	↑	↑	↑						
	mid	low	high						

mid

$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

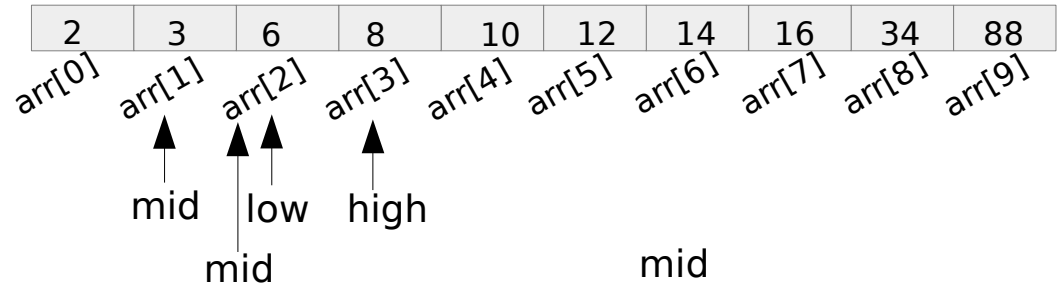
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

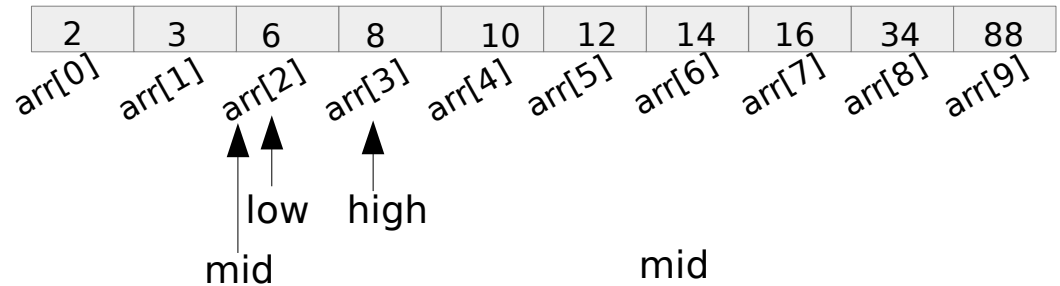
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

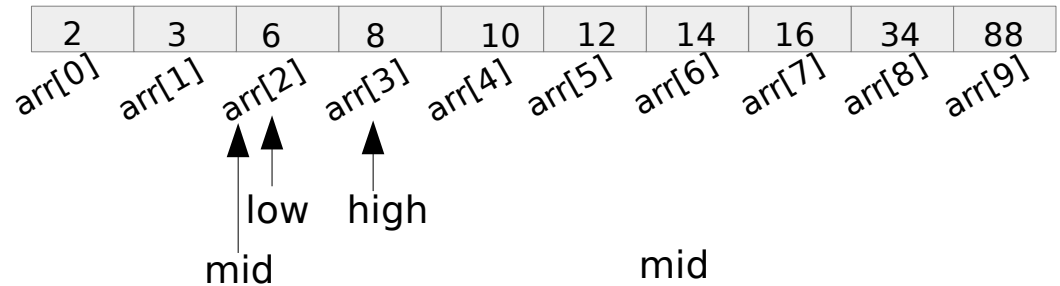
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

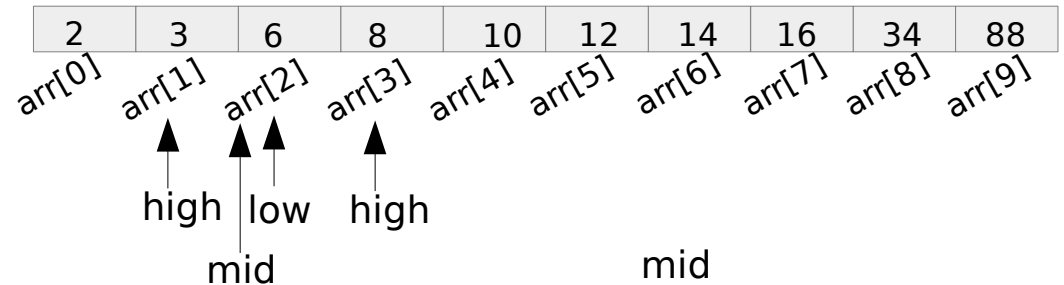
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

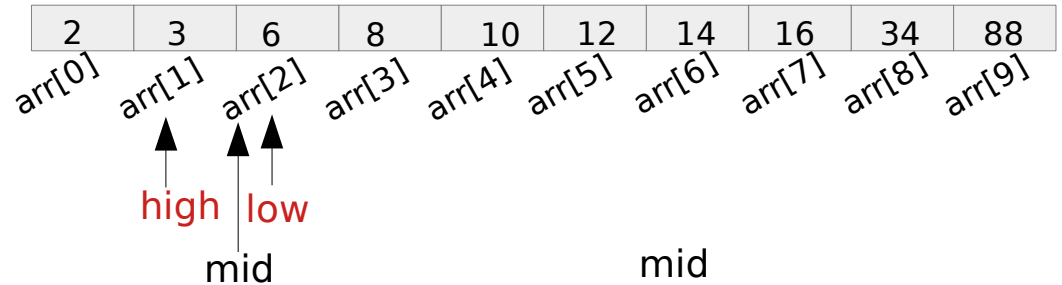
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

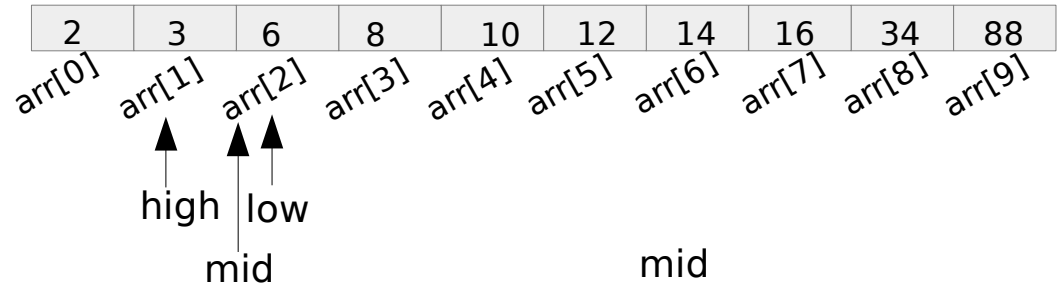
# Binary\_search\_iterative(arr,key)

```
low = 0 ,high = size-1
while ( low <= high )
    mid = (low+high) / 2
    if (arr[mid] == key )
        return mid
    else if (key < a[mid])
        high = mid - 1
    else
        low = mid + 1
return -1
```

arr[SIZE]

SIZE = 10

key=5



$$(0+9)/2 = 4$$

$$(0+3)/2 = 1$$

$$(2+3)/2 = 2$$

# Code - Binary Search Iterative