Searching

Search 33

Search for key = 33 in the following sorted array

6	13	14	25	33	43	51	53	64	72	84

Linear Search

```
for(int i=0; i < length(a); i++){

    if(a[i]== key) return i;

}

return -1;
```

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Search 33

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Search 33

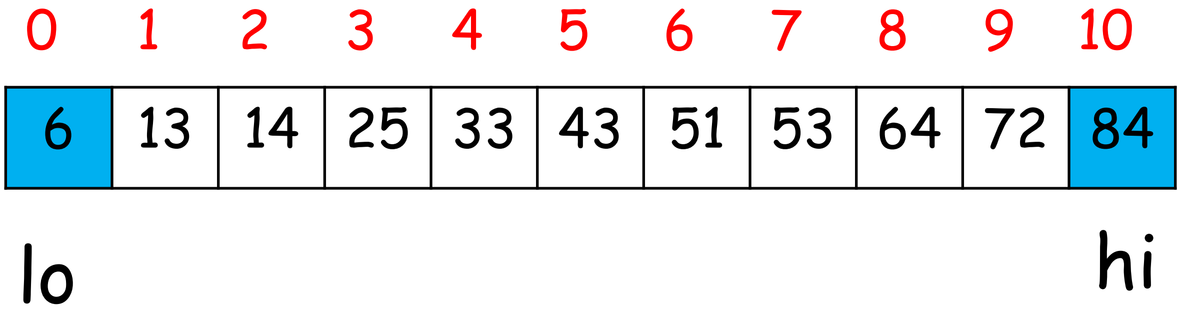
0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

33

Binary Search

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi - lo) / 2;
    if(key < a[mid]) hi = mid - 1;
    else if (key > a[mid]) lo = mid + 1;
    else return mid;
}
```

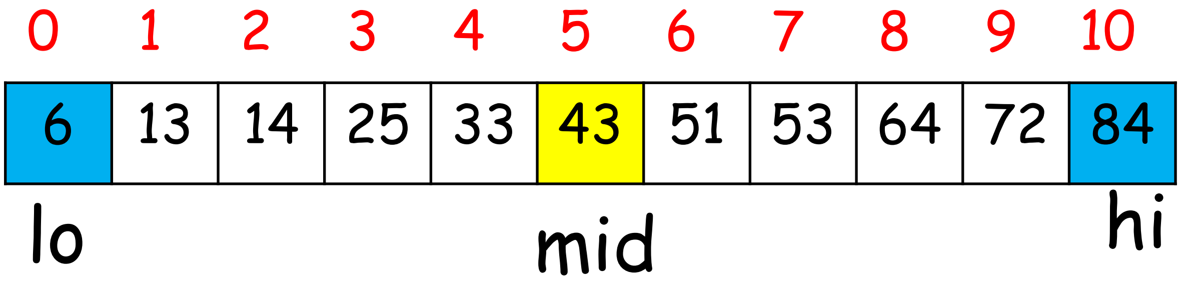
Search 33



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

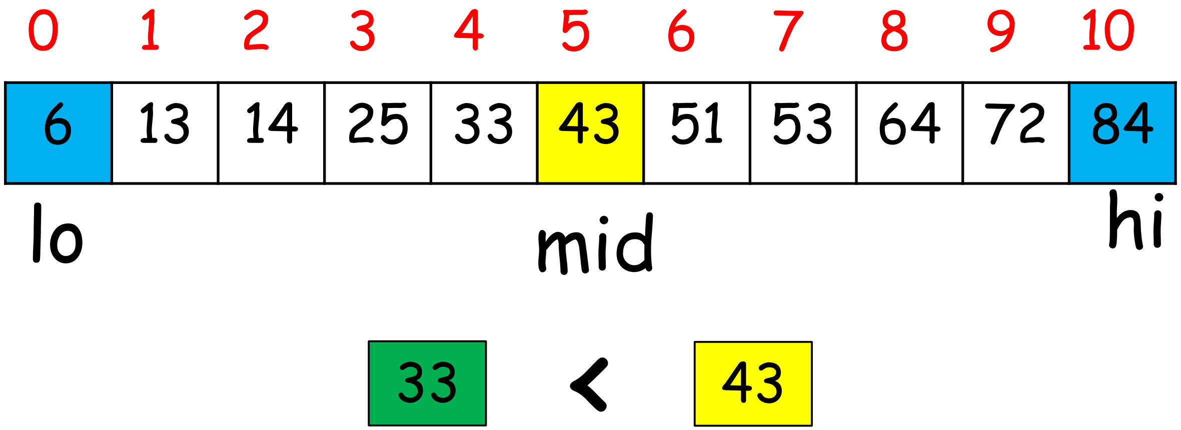
Search 33:1st iteration



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
        lo) / 2;
    if(key < a[mid])
        hi = mid - 1;
    else if (key > a[mid])
        lo = mid + 1;
    else
        return mid;
}
```

Search 33:1st iteration



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

33 < 43

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
            lo) / 2;
    if(key < a[mid])
            hi = mid - 1;
    else if (key > a[mid])
            lo = mid + 1;
    else
            return mid;
}
```

Search 33:1st iteration

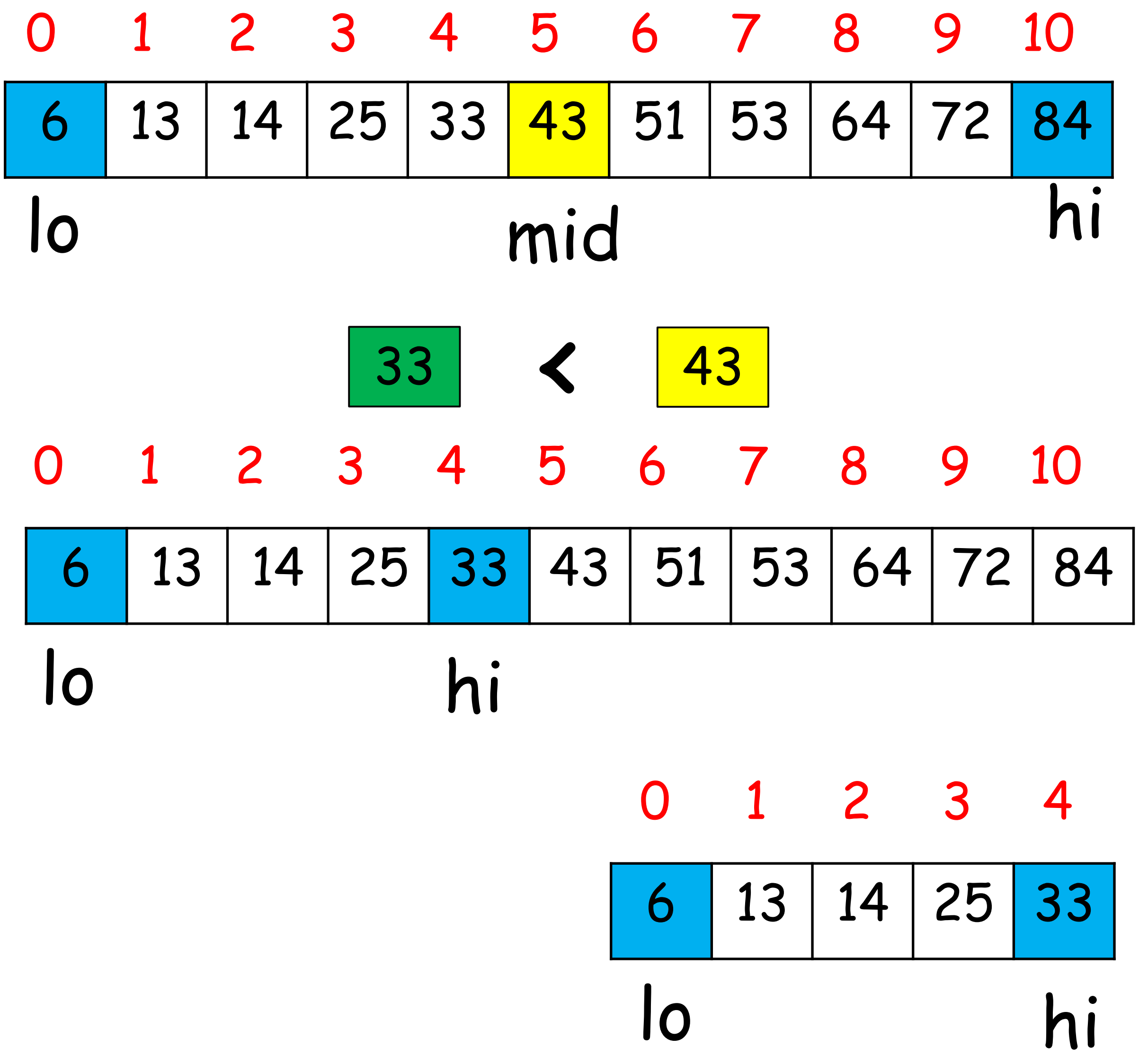
0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

33 < 43

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo				hi						

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:1st iteration



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

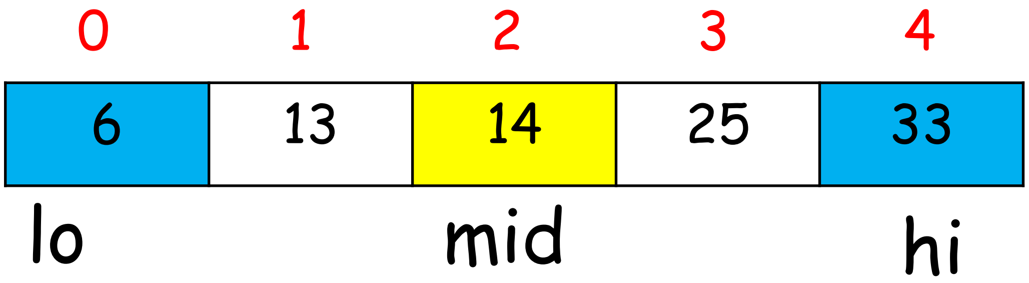
33 < 43

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo				hi						

0	1	2	3	4
6	13	14	25	33
lo				hi

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

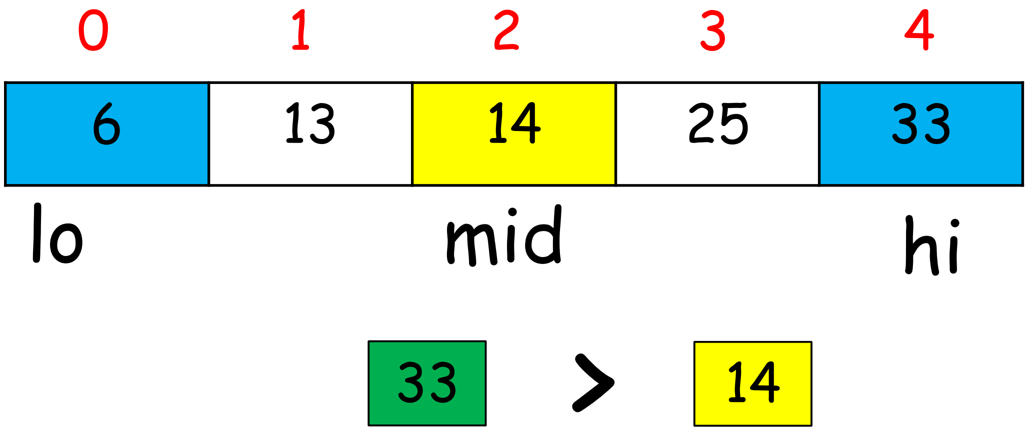
Search 33:2nd iteration



0	1	2	3	4
6	13	14	25	33
lo		mid		hi

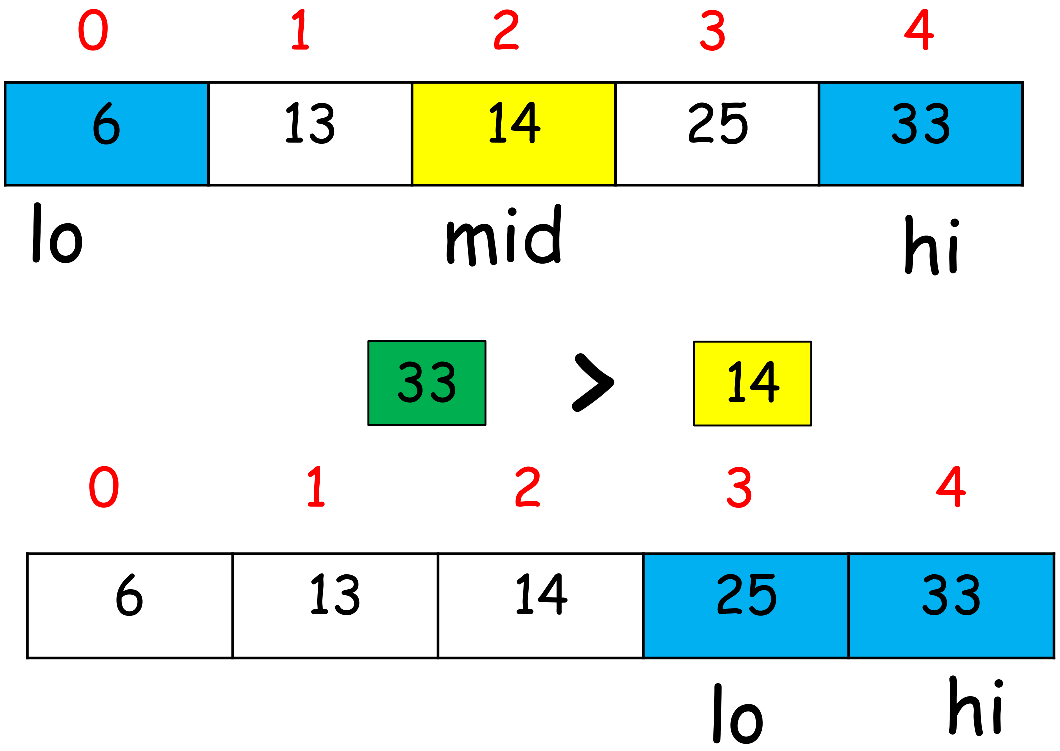
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
          lo) / 2;
    if(key < a[mid])
          hi = mid - 1;
    else if (key > a[mid])
          lo = mid + 1;
    else
          return mid;
}
```

Search 33:2nd iteration



```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
        int mid = lo + (hi -
                  lo) / 2;
        if(key < a[mid])
                  hi = mid - 1;
        else if (key > a[mid])
                  lo = mid + 1;
        else
                  return mid;
}
```

Search 33:2nd iteration



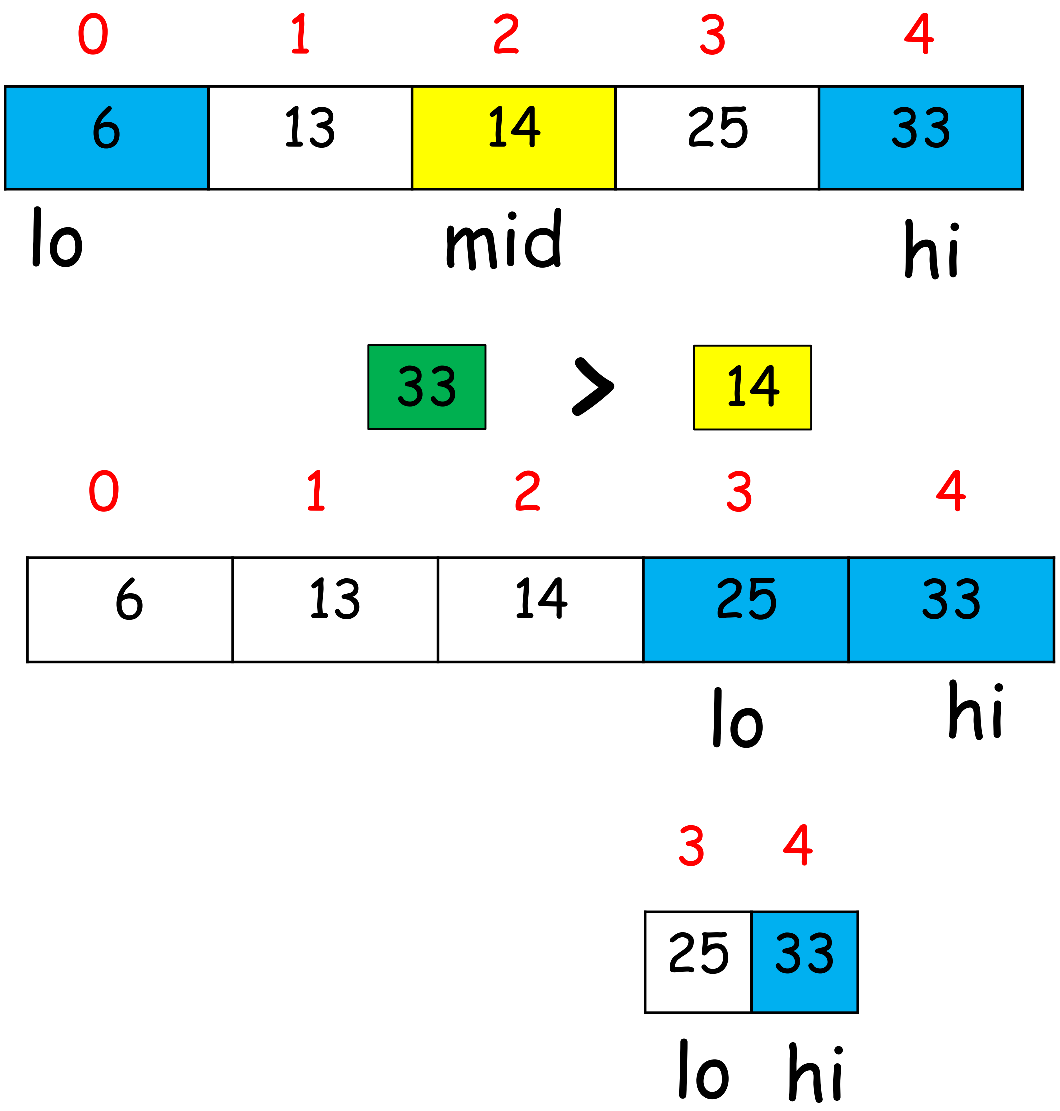
0	1	2	3	4
6	13	14	25	33
lo		mid		hi

33 > 14

0	1	2	3	4
6	13	14	25	33
			lo	hi

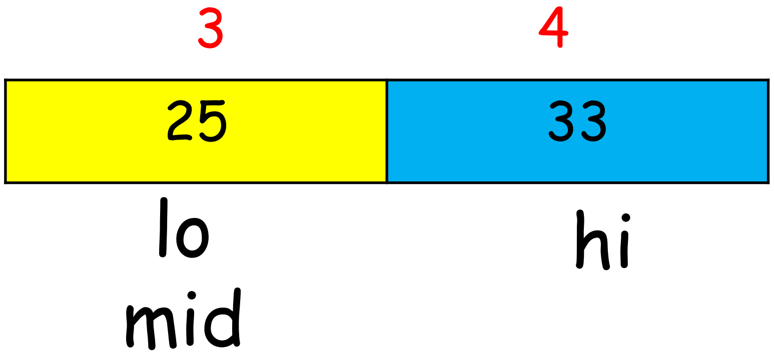
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:2nd iteration



```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
          lo) / 2;
    if(key < a[mid])
          hi = mid - 1;
    else if (key > a[mid])
          lo = mid + 1;
    else
          return mid;
}
```

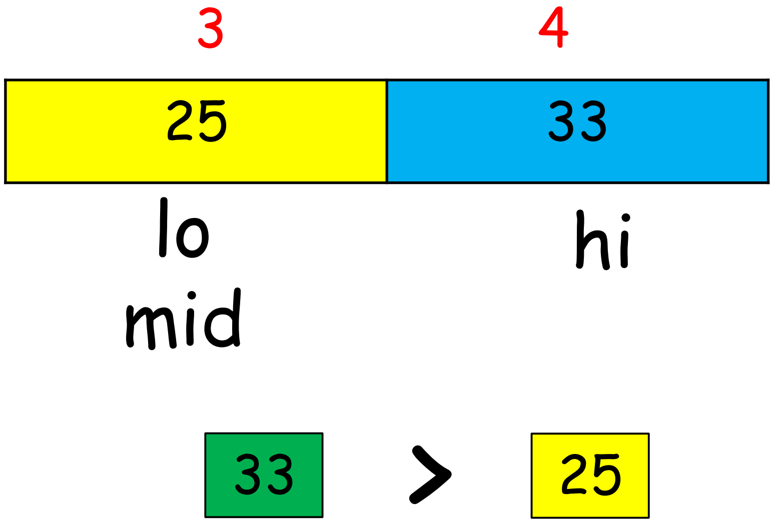
Search 33:3rd iteration



3	4
25	33
lo mid	hi

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:3rd iteration



3

4

25

33

lo

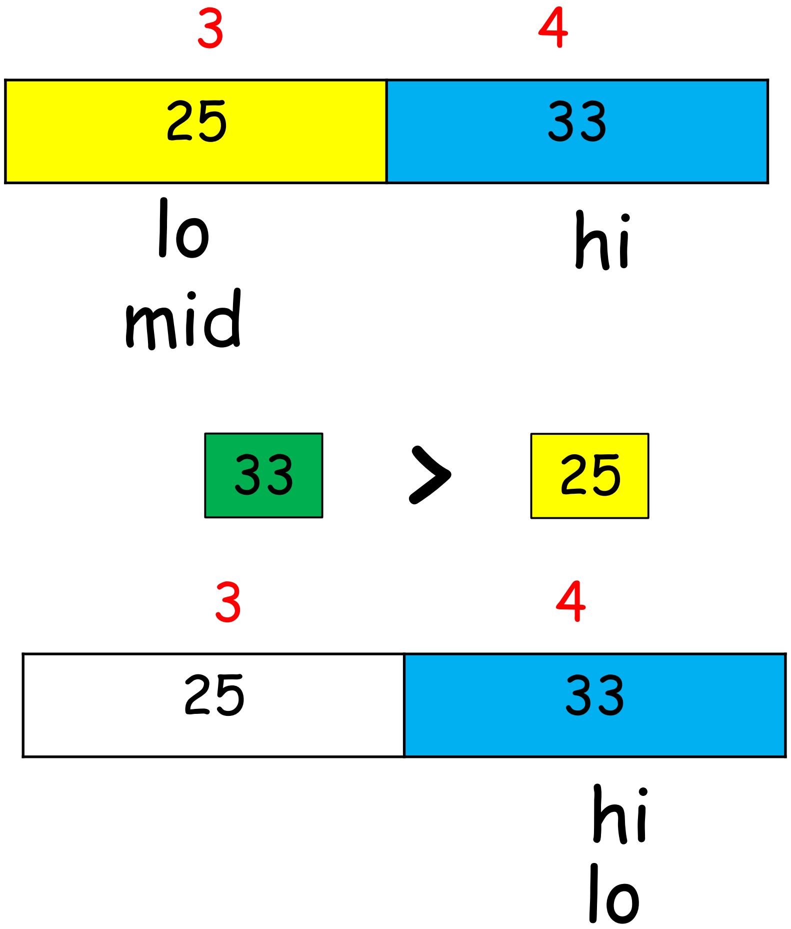
mid

hi

33 > 25

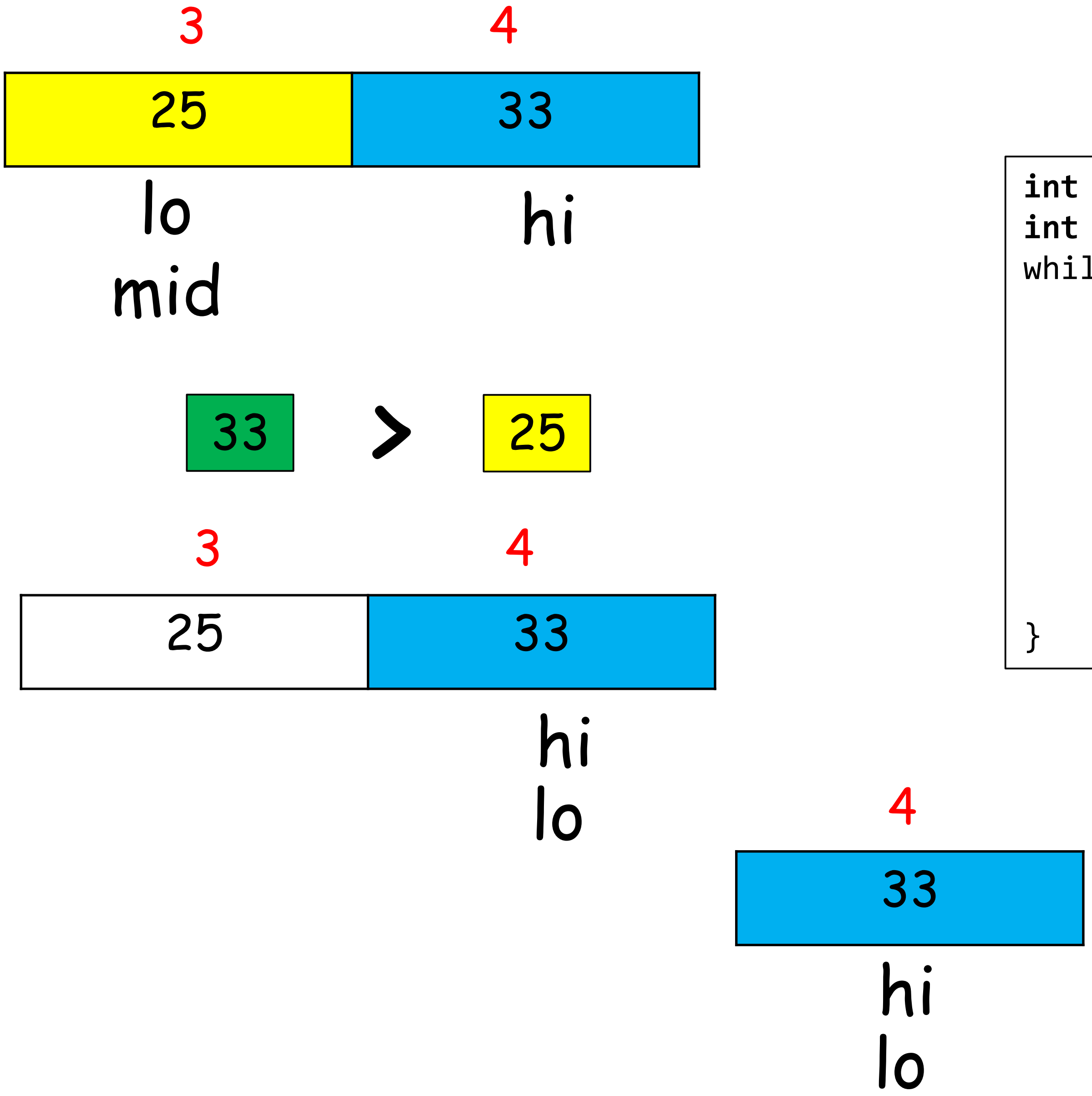
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:3rd iteration



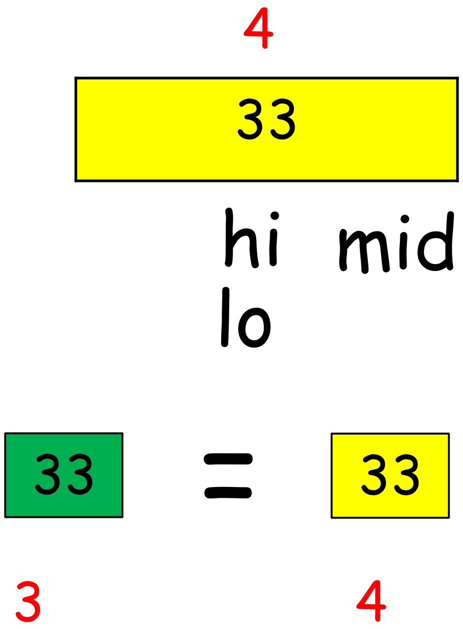
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:3rd iteration



```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
        lo) / 2;
    if(key < a[mid])
        hi = mid - 1;
    else if (key > a[mid])
        lo = mid + 1;
    else
        return mid;
}
```

Search 33:4th iteration



4

33

hi mid
lo

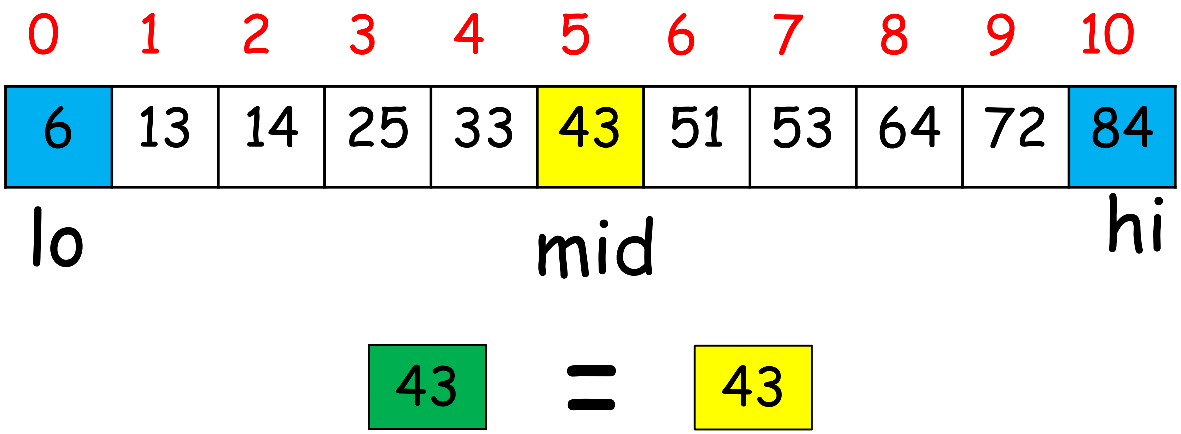
33 = 33

3 4

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
          lo) / 2;
    if(key < a[mid])
          hi = mid - 1;
    else if (key > a[mid])
          lo = mid + 1;
    else
          return mid;
}
```

Output: 4

Search 43:1st iteration

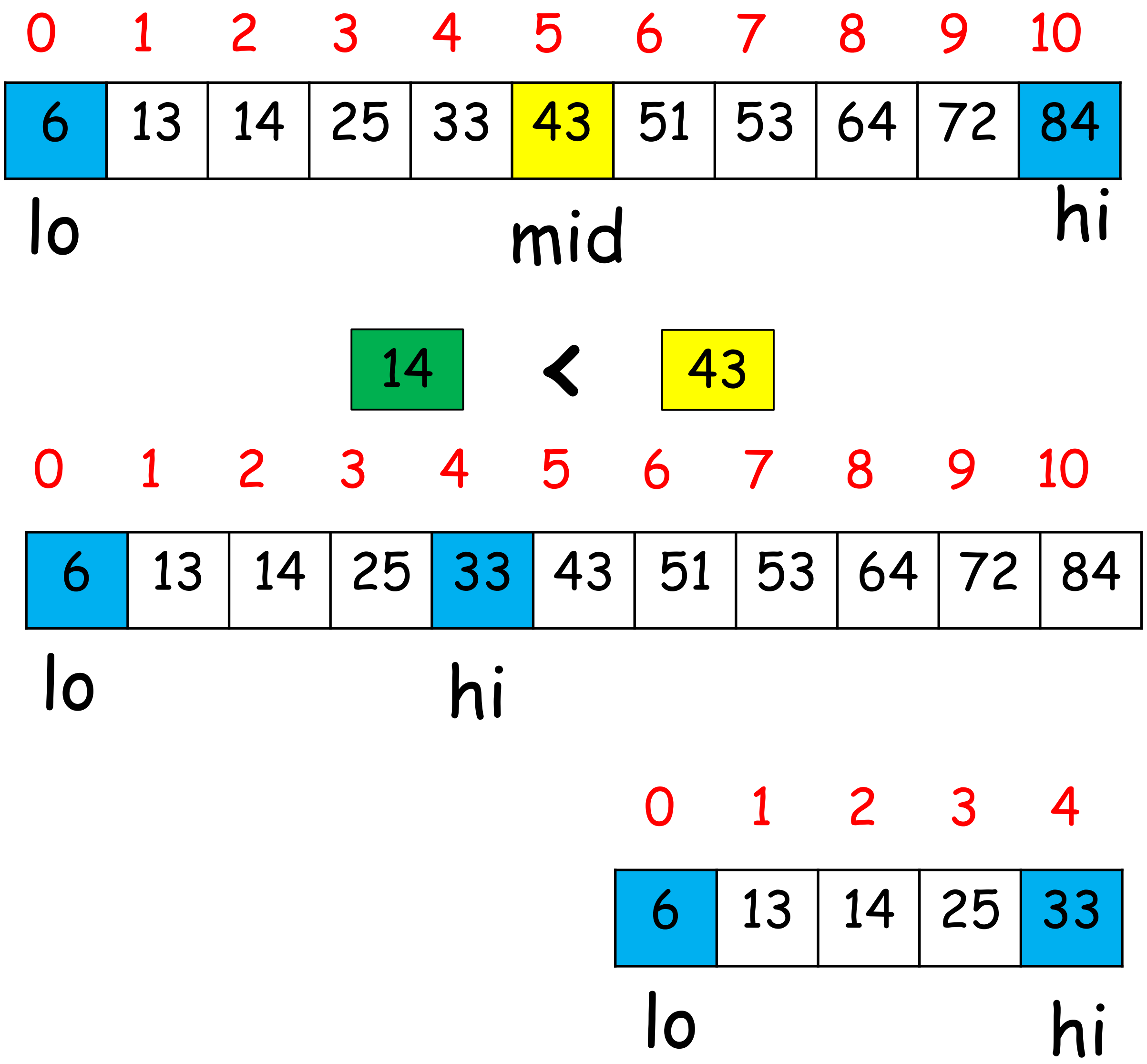


0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Output = 5

Search 14:1st iteration



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

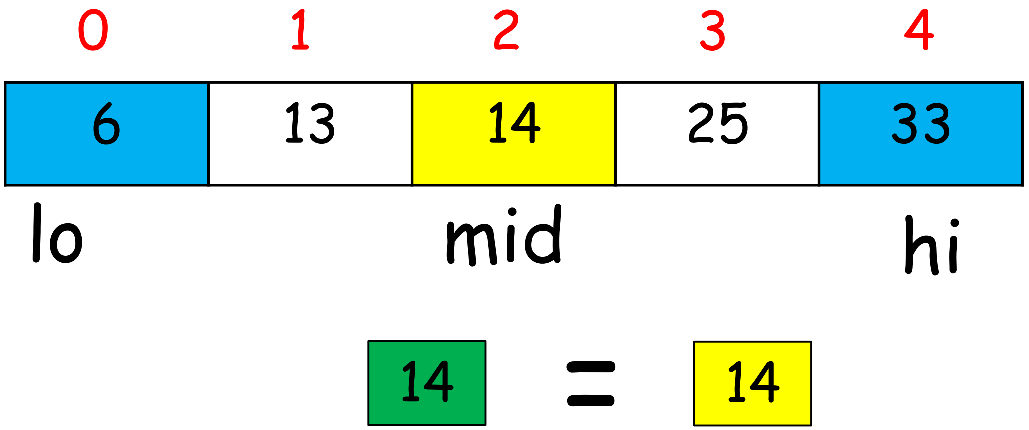
14 < 43

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo				hi						

0	1	2	3	4
6	13	14	25	33
lo				hi

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 14:2nd iteration



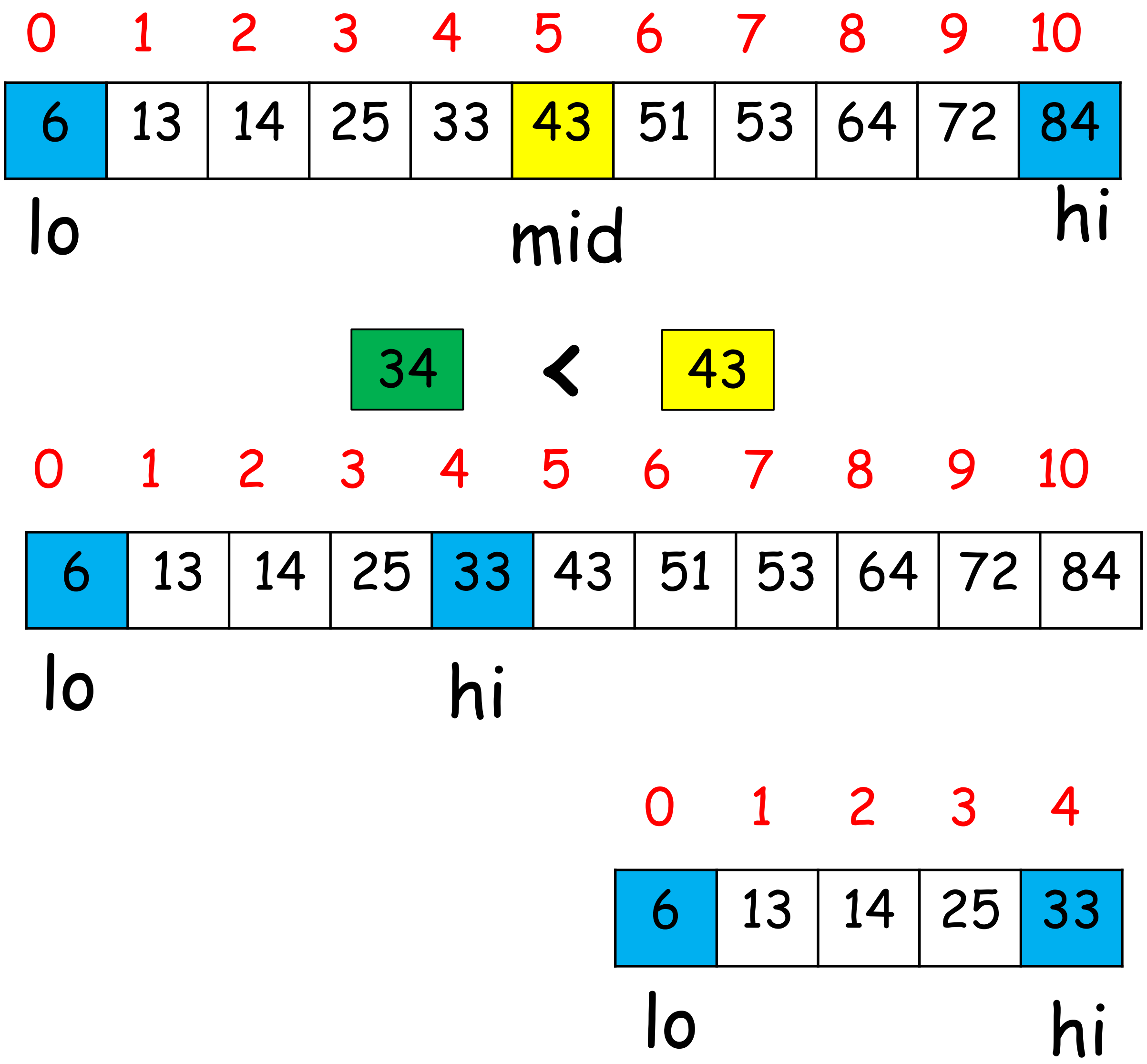
0	1	2	3	4
6	13	14	25	33
lo		mid		hi

14 = 14

Output = 2

```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
          lo) / 2;
    if(key < a[mid])
          hi = mid - 1;
    else if (key > a[mid])
          lo = mid + 1;
    else
          return mid;
}
```

Search 34:1st iteration



0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo					mid					hi

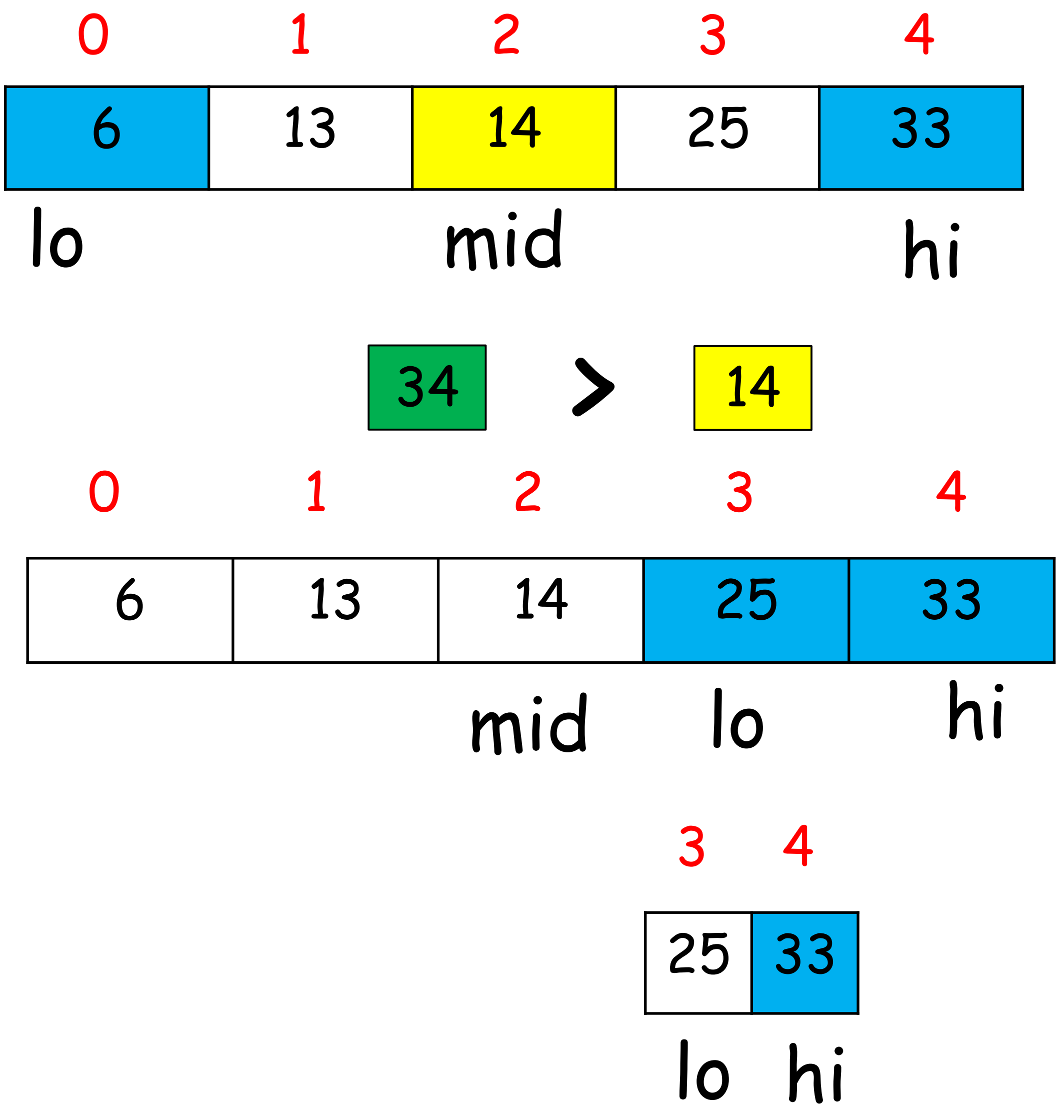
34 < 43

0	1	2	3	4	5	6	7	8	9	10
6	13	14	25	33	43	51	53	64	72	84
lo				hi						

0	1	2	3	4
6	13	14	25	33
lo				hi

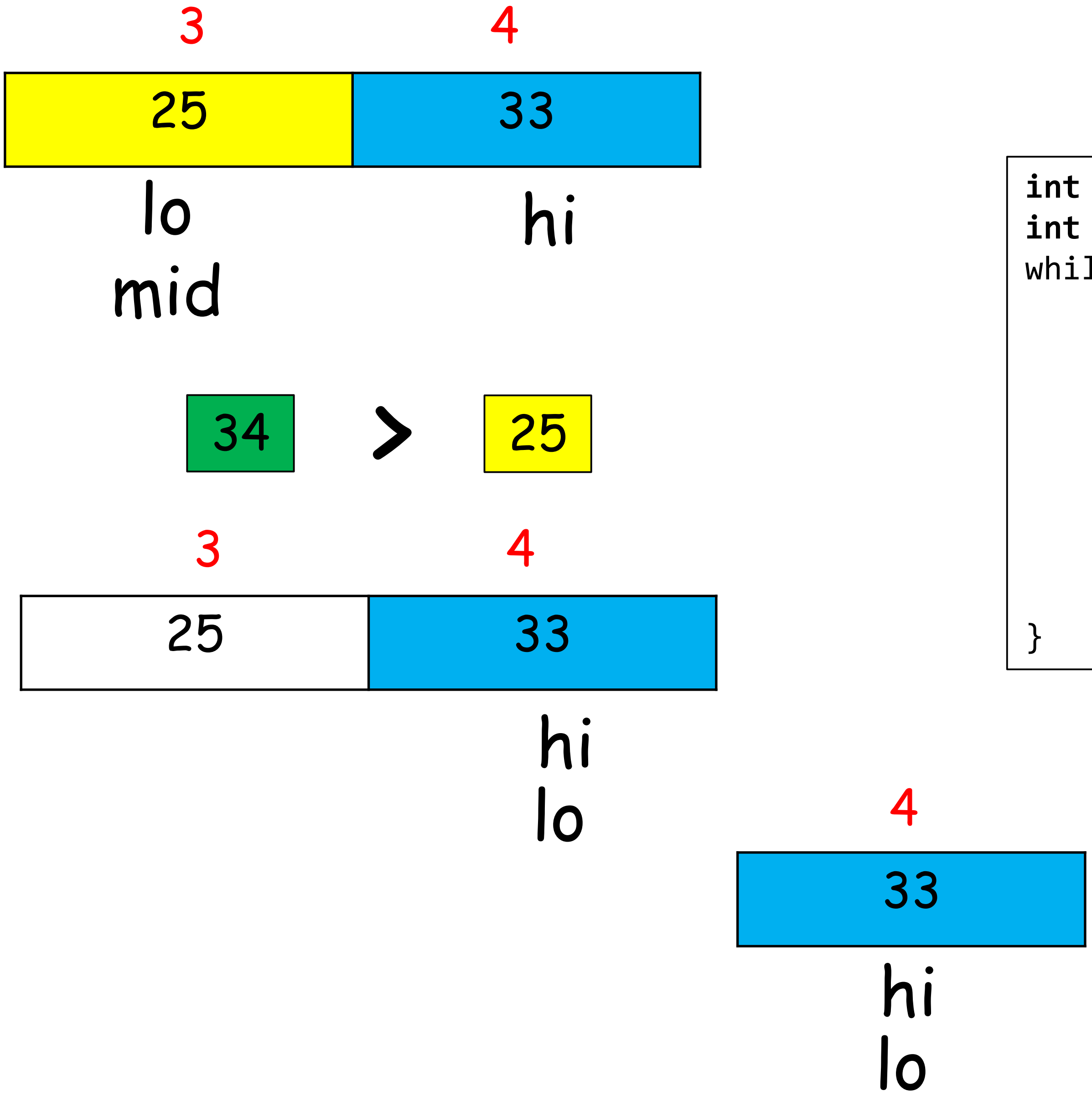
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:2nd iteration



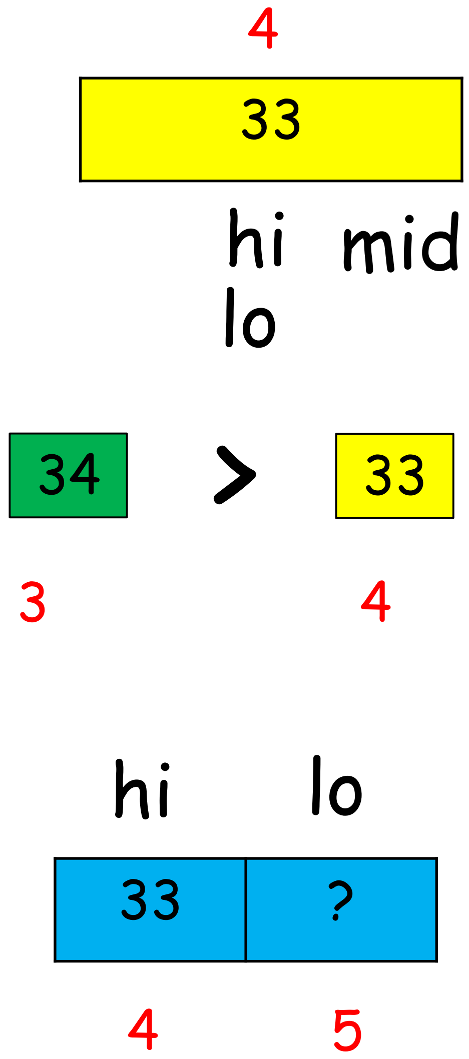
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:3rd iteration



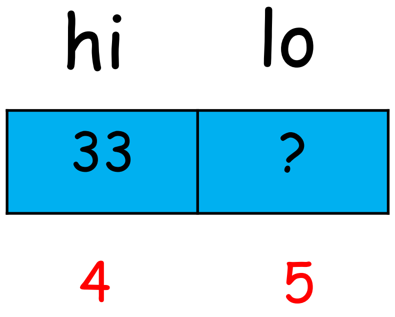
```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
         lo) / 2;
    if(key < a[mid])
         hi = mid - 1;
    else if (key > a[mid])
         lo = mid + 1;
    else
         return mid;
}
```

Search 33:4th iteration



```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
        lo) / 2;
    if(key < a[mid])
        hi = mid - 1;
    else if (key > a[mid])
        lo = mid + 1;
    else
        return mid;
}
```

Search 33:4th iteration



```
int lo = 0;
int hi = array_size - 1;
while (lo <= hi) {
    int mid = lo + (hi -
        lo) / 2;
    if(key < a[mid])
        hi = mid - 1;
    else if (key > a[mid])
        lo = mid + 1;
    else
        return mid;
}
```

Output = -1