Insertion Sort Demo

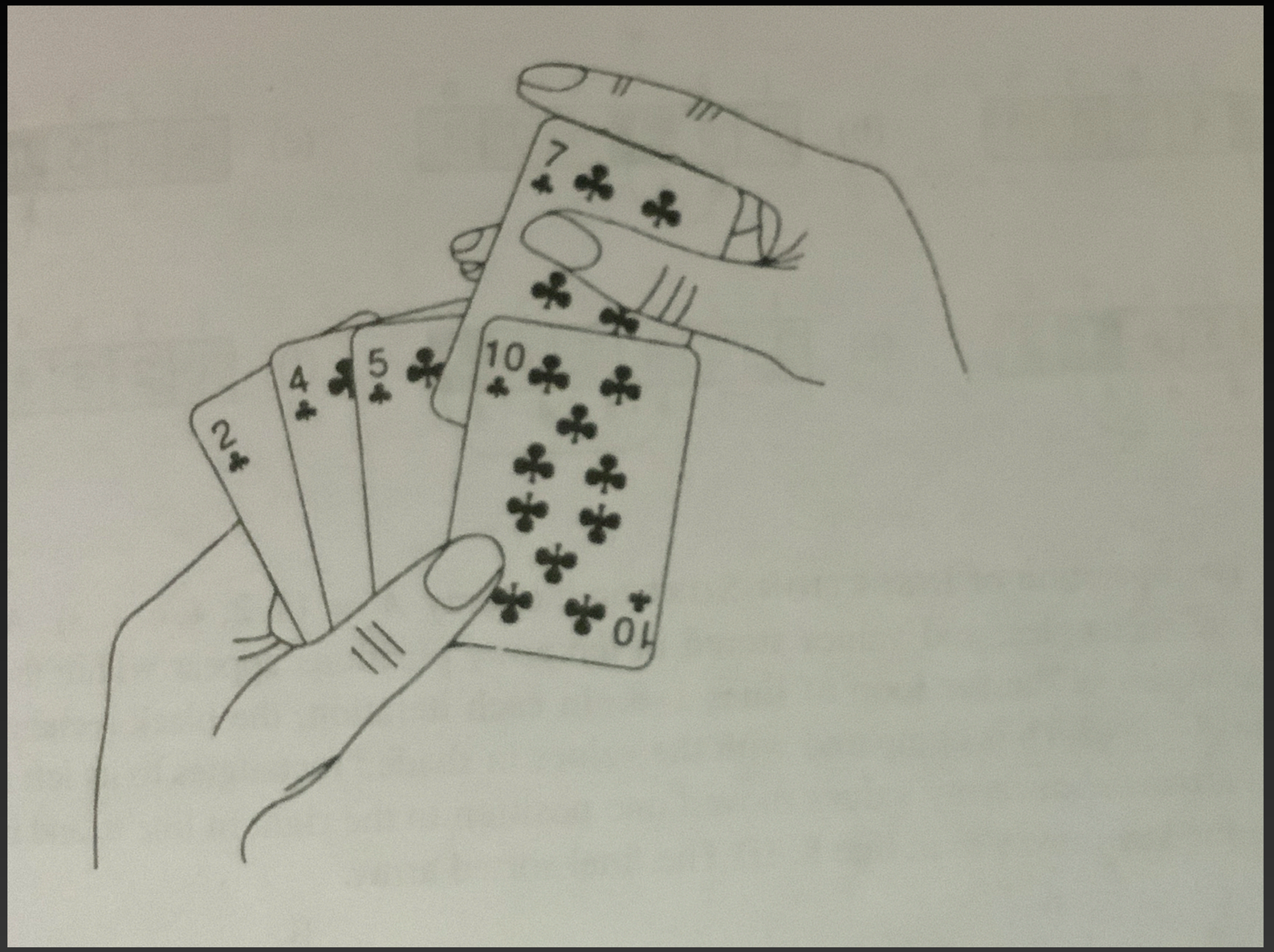
Refer to chapter 2 from Cormen

Sakeena Shahid 14-01-2022

Insertion Sort

An algorithm to sort an input sequence in-place

- Input: A sequence of numbers stored <a1, a2, … , an>
- Output: a reordering (permutation) <a1’, a2’, … , an’> such that a1’ <= a2’ <= a3’ …. <= an’
- The individual elements of the input sequence are called **keys**
- The input generally comes to us as an array with n elements
- So, **Input :** Array: A[1], A[2], … , A[n]
- **Output:** Sorted array: A[1]<=A[2]<= … <= A[n]


7
2
4
5
10

Implementation of Insertion Sort

Input : Array: A[1], A[2], … , A[n]

Output: Sorted array: A[1]<=A[2]<= … <= A[n]

```
for j = 2 to n do
    key = A[j]
    i = j - 1
    while I > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i - 1
    end
    A[i + 1] = key
end
```

Example

Array A

Fig 1: Array to be sorted

Example

Array A

i = 1

j = 2

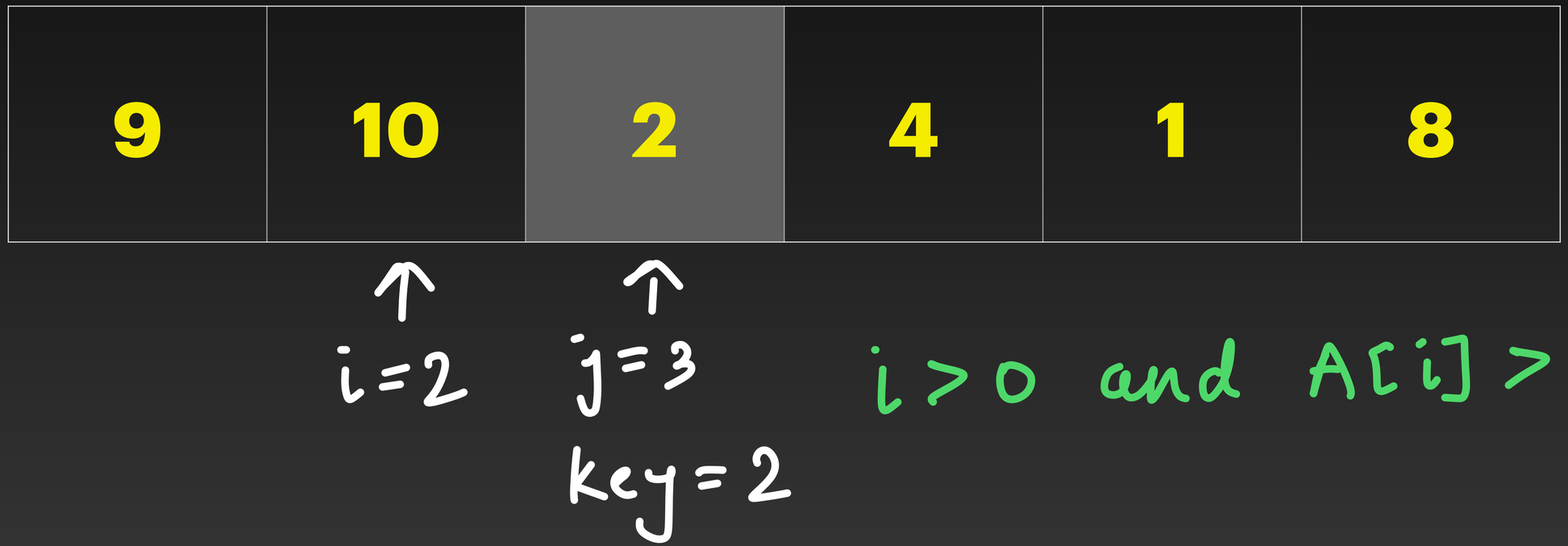
key = 10

i > 0 and A[i] > key?

Iteration 1 : Step 1

Example

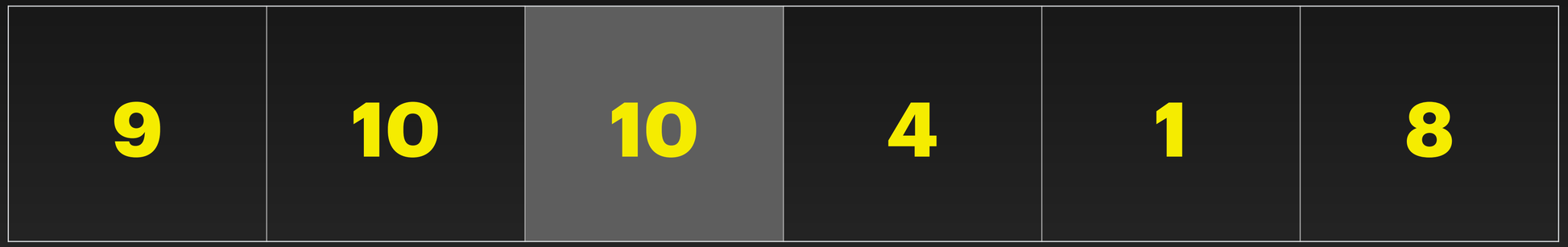
Array A



Iteration 2: Step 1

Example

Array A



i = 1

j = 3

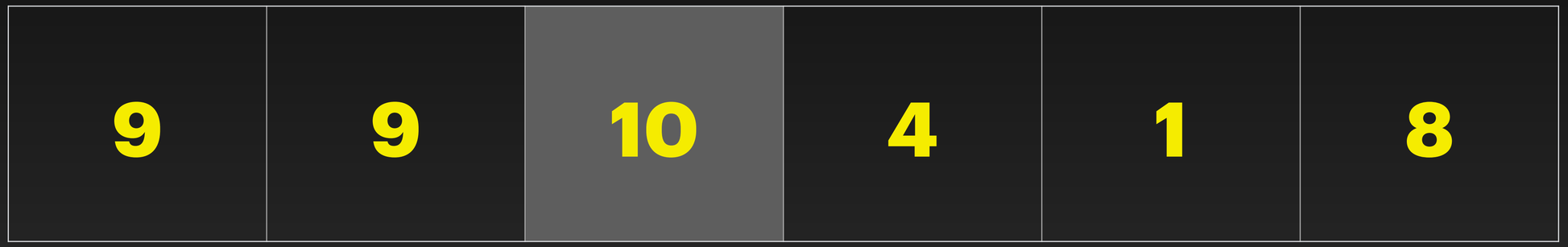
key = 2

i > 0 and A[i] > key?

Iteration 2: Step 2

Example

Array A



i = 0

j = 3

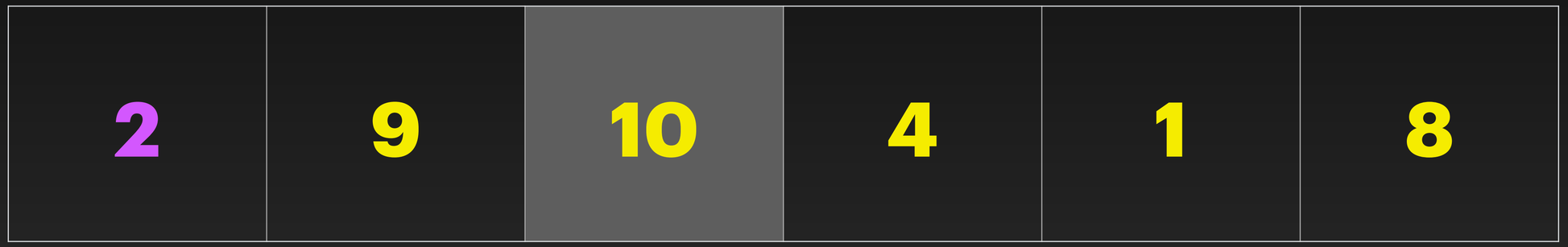
key = 2

i > 0 and A[i] > key?

Iteration 2: Step 3

Example

Array A



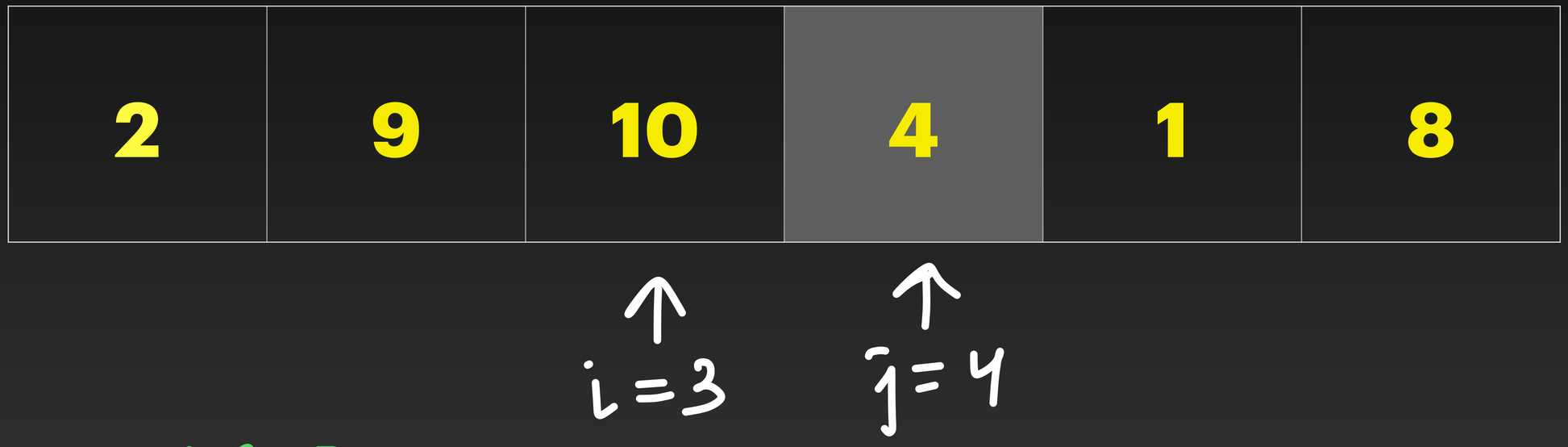
A[i+1] = key

key = 2

Iteration 2: Step 4

Example

Array A

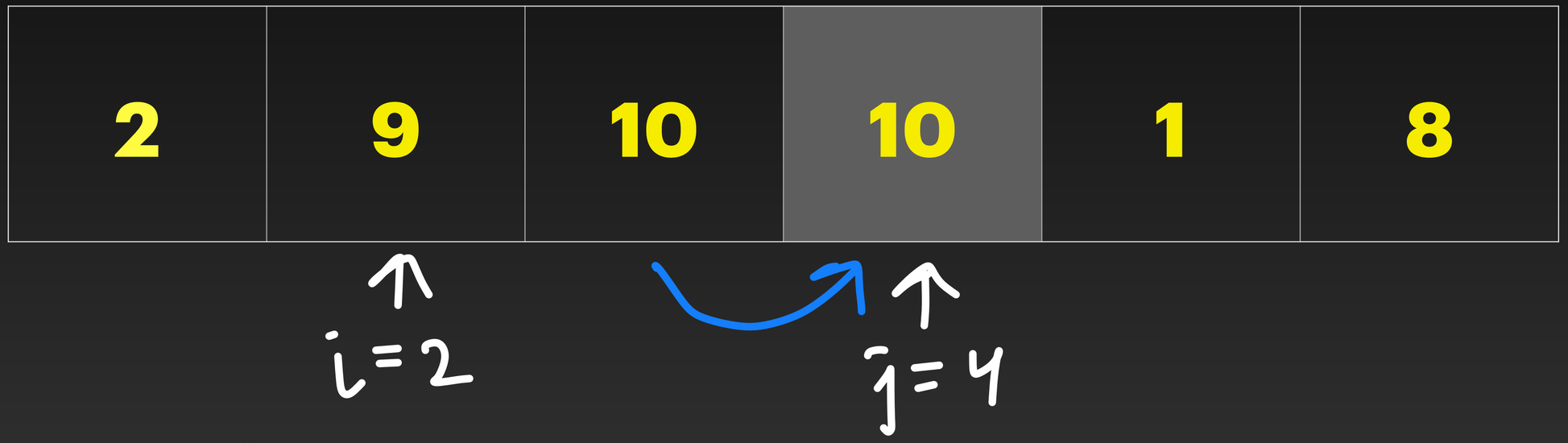


i > 0 and A[i] > key?

Iteration 3: Step 1

Example

Array A



i > 0 and A[i] > key?

Iteration 3: Step 2

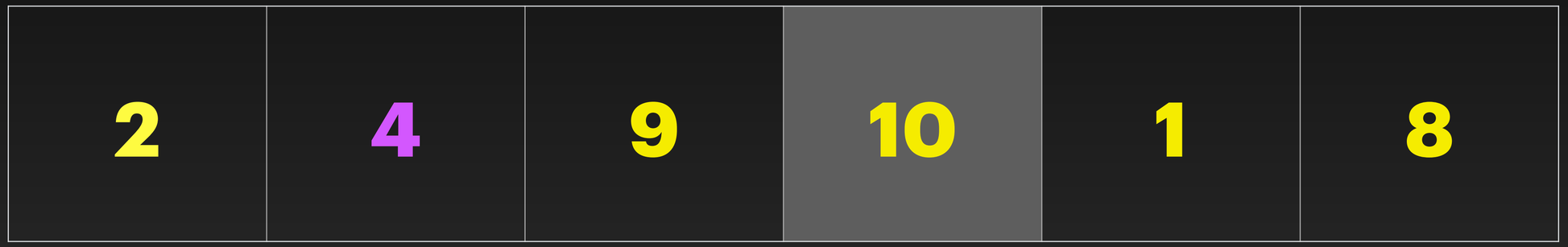
Example

Array A

Iteration 3: Step 3

Example

Array A



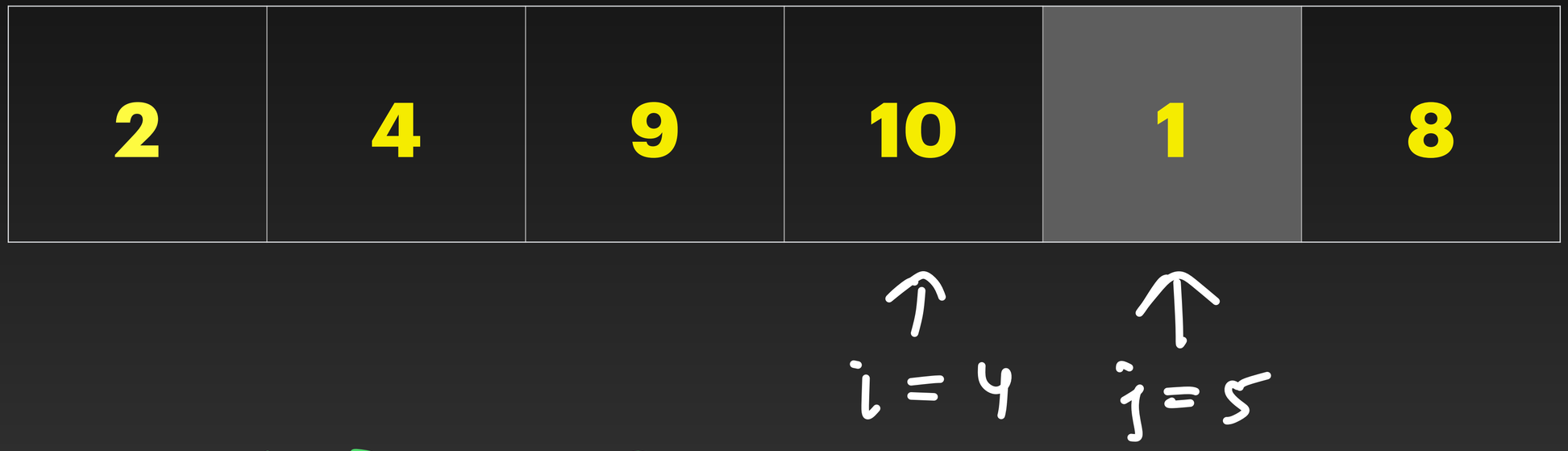
A[i+1] = key

key = 4

Iteration 3: Step 4

Example

Array A



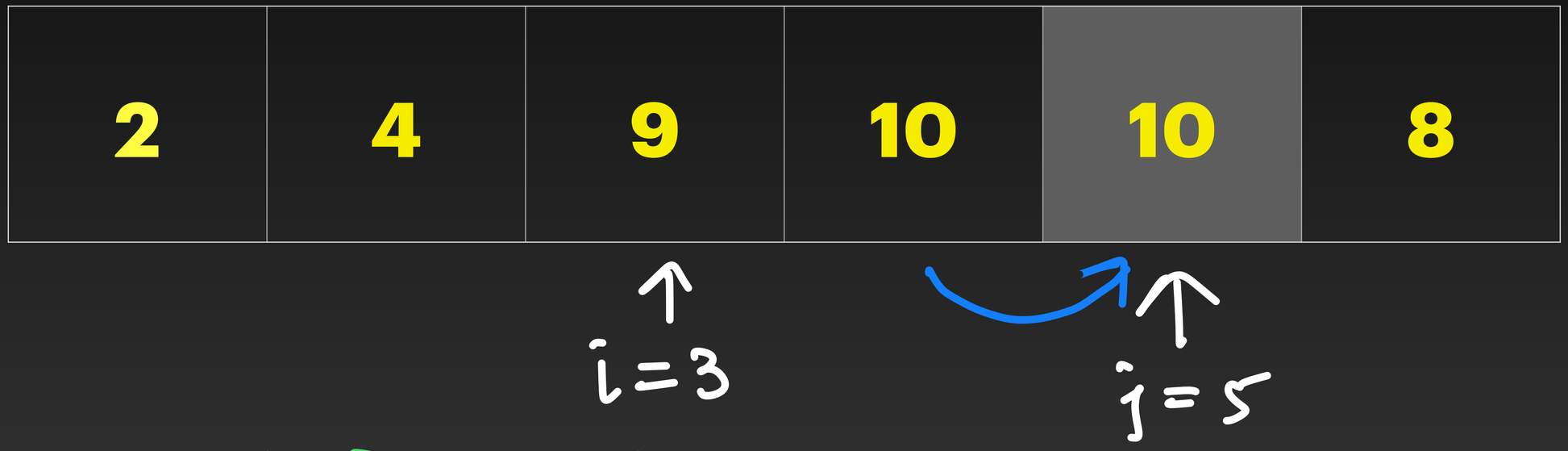
i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 1

Example

Array A



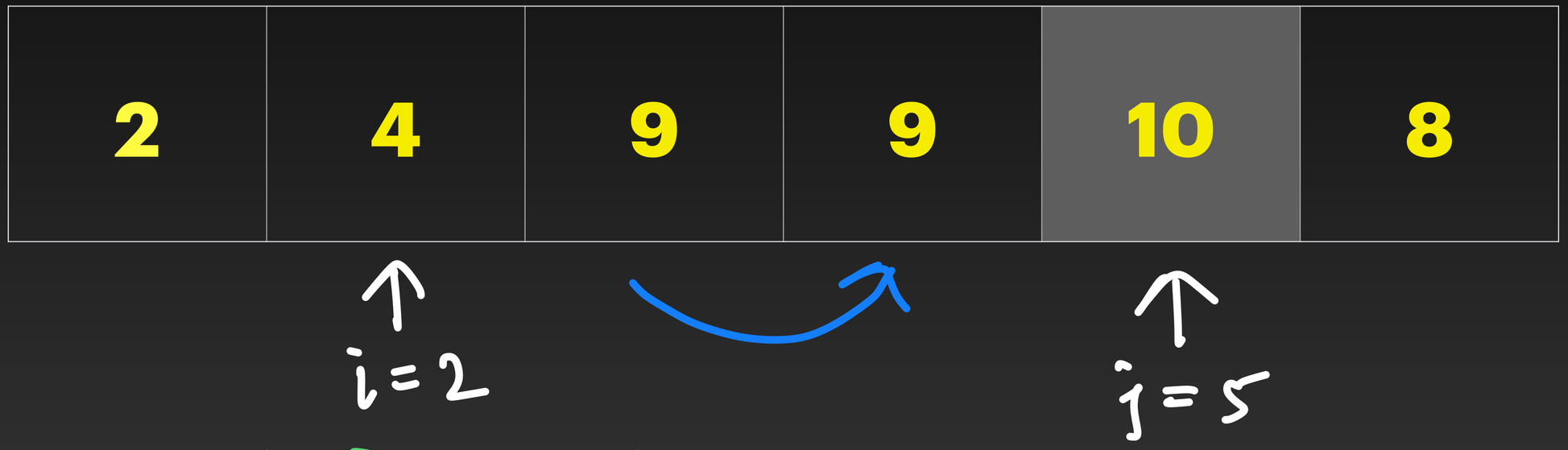
i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 2

Example

Array A



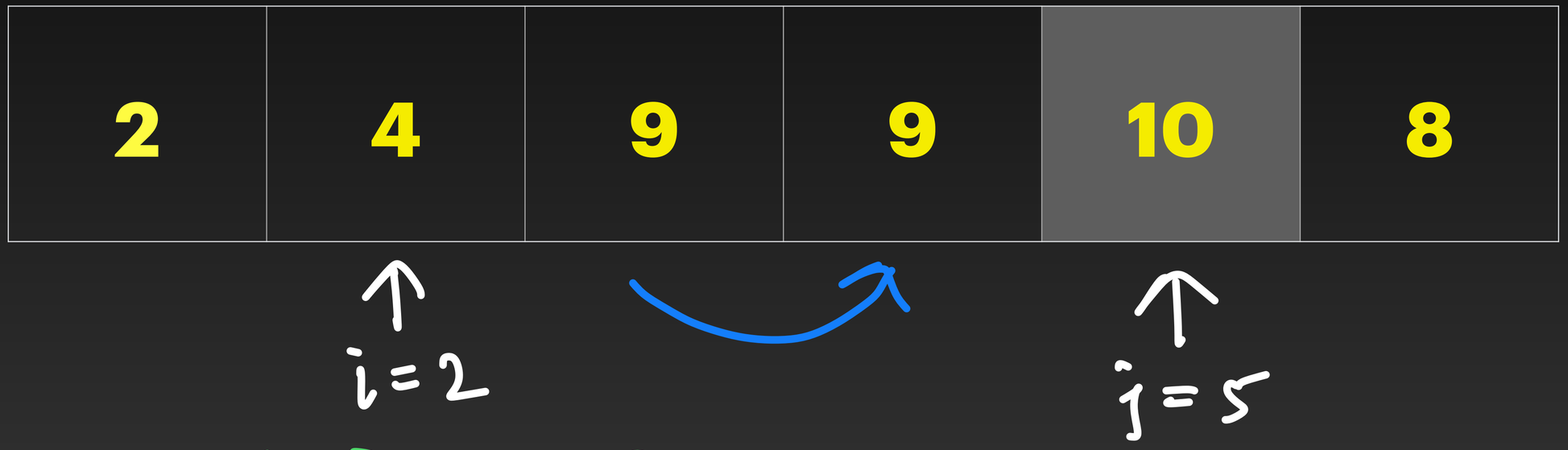
i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 3

Example

Array A



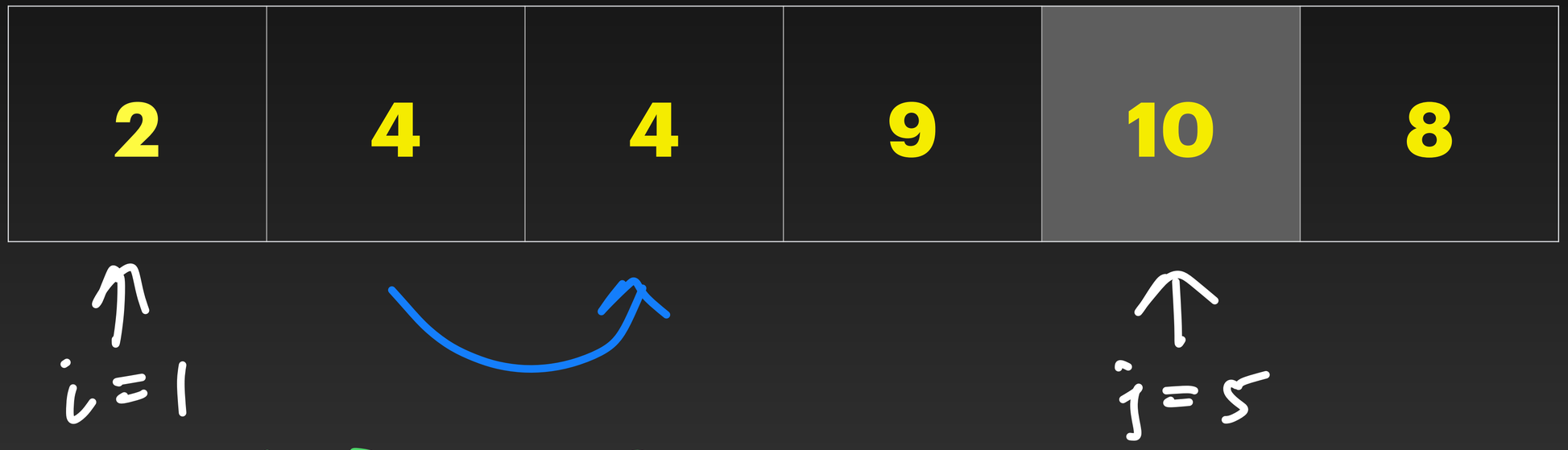
i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 3

Example

Array A



i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 4

Example

Array A

i = 0

j = 5

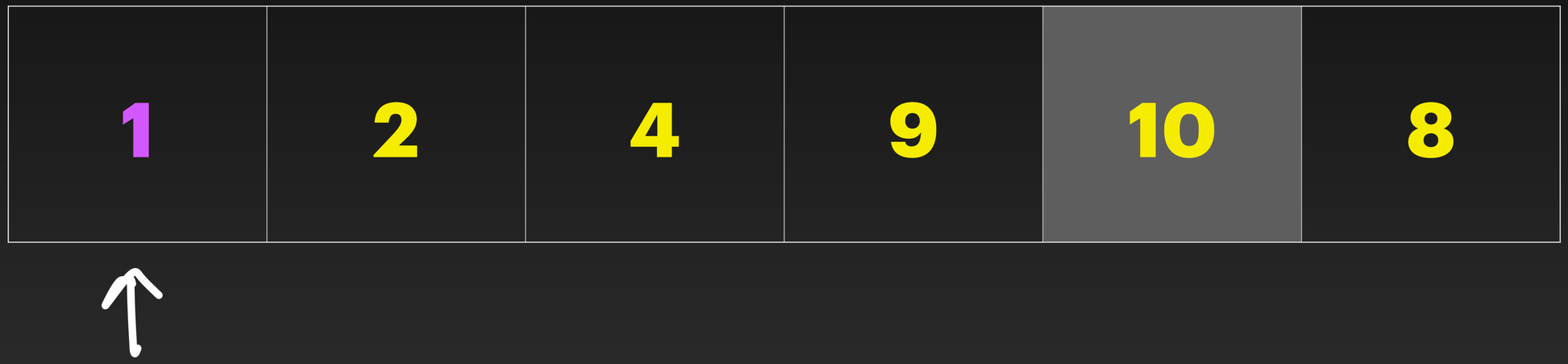
i > 0 and A[i] > key?

key = 1

Iteration 4 : Step 5

Example

Array A



A[i+1] = key

key = 1

Iteration 4 : Step 6

Example

Array A

i = 5

j = 6

key = 8

i > 0 and A[i] > key?

Iteration 5 : Step 1

Example

Array A

i = 4

j = 6

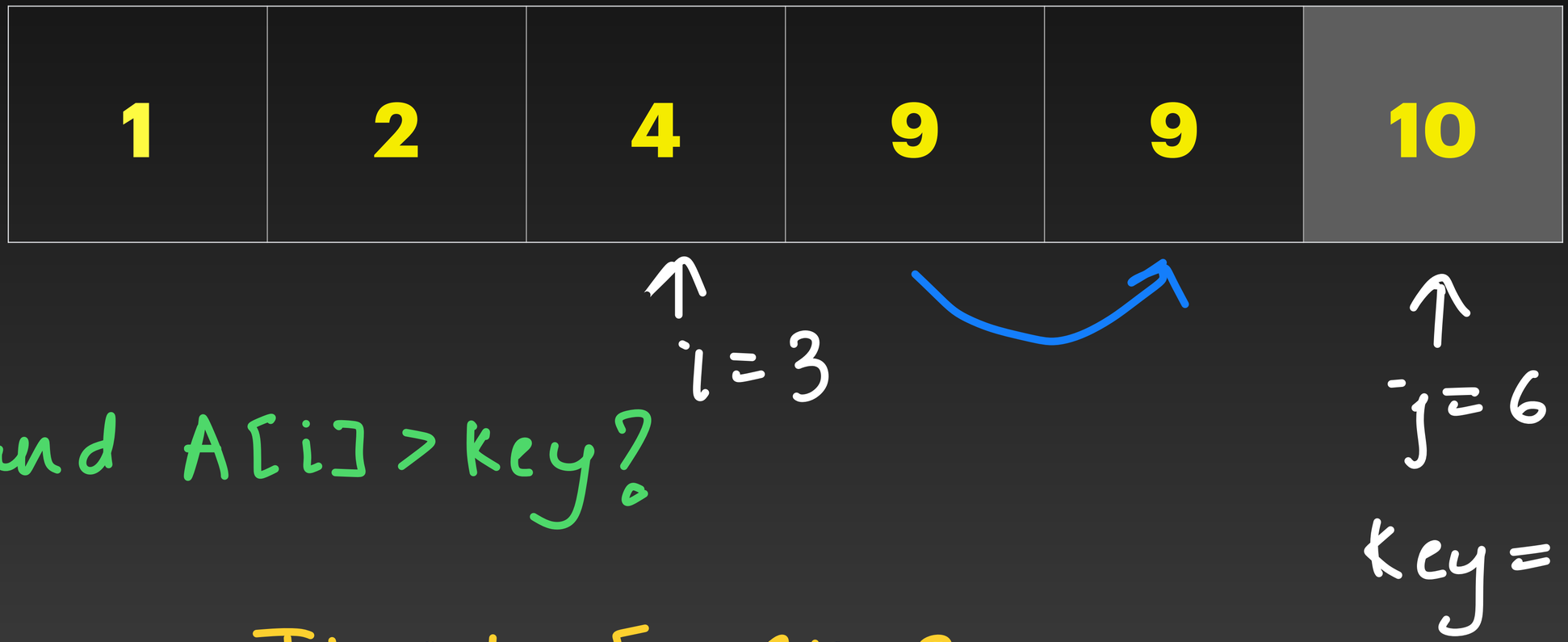
key = 8

i > 0 and A[i] > key?

Iteration 5 : Step 2

Example

Array A

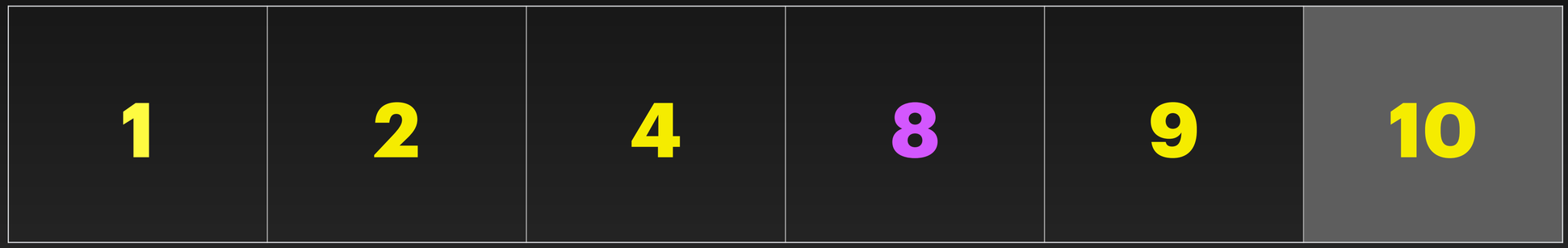


i > 0 and A[i] > key?

Iteration 5 : Step 2

Example

Array A



A[i+1] = key

key = 8

Iteration 5 : Step 3

Example

Array A

Fig 2 : Array after sorting

What will be the best case?

sorted array

not going through the while loop even once.

Example

1 comparison
0 movements

Array A

Best case : All elements are already sorted

Fig 3: Best case of IS

Example

2 comparison
0 movements

Array A

1	2	4	8	9	10

i = 2 j = 3

Best case : All elements are already sorted

Fig 3: Best case of IS

Example

3 comparison
0 movements

Array A

1	2	4	8	9	10

i=3 ↑ ↑ j=4

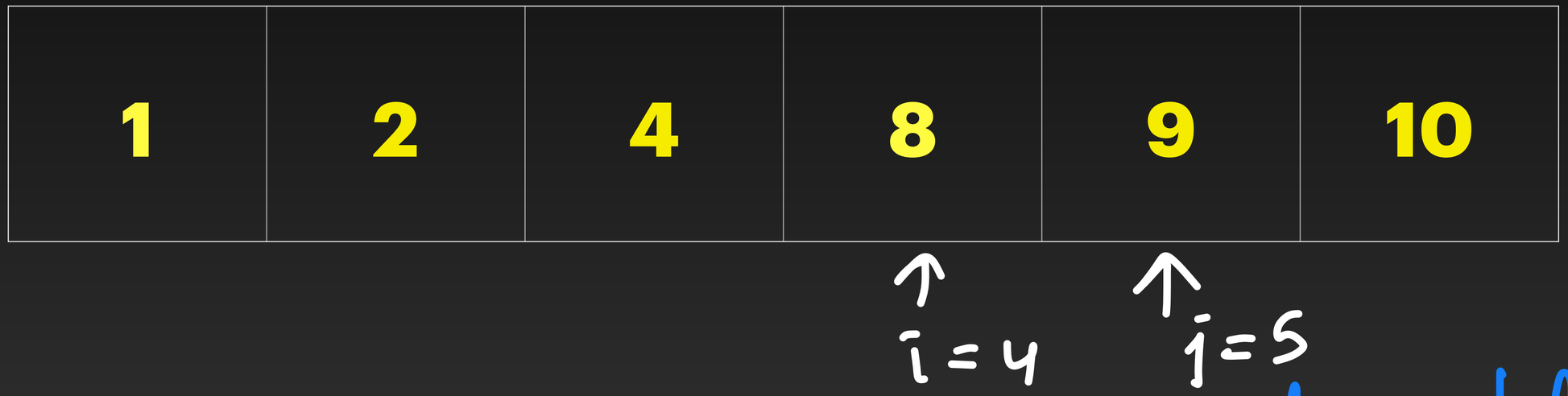
Best case : All elements are already sorted

Fig 3: Best case of IS

Example

4 comparison
0 movements

Array A



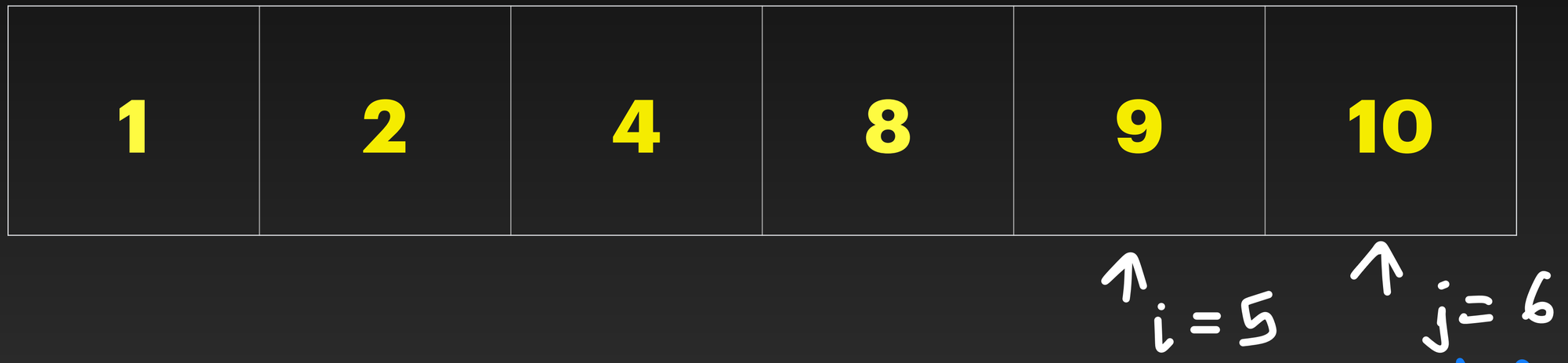
Best case : All elements are already sorted

Fig 3: Best case of IS

Example

5 comparison
0 movements

Array A



Best case : All elements are already sorted

Fig 3: Best case of IS

What will be the worst case?

↳ reverse sorted array

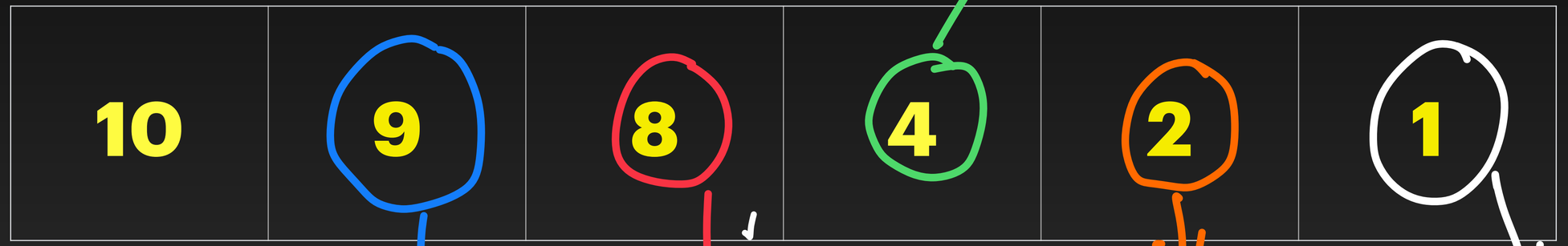
↓

for EVERY element, go through the while loop.

Fig 3: Worst case of IS

Example

Array A



key { 3 comparisons + 3 movements

key

key

2 comparisons + 2 movements

key

1 comparison + 1 movement

1+2+3+...+(n-1) comparisons + 1+2+3+....+(n-1) movements

n(n-1)/2 C

n(n-1)/2 M

Insertion Sort.

Worst case	$O(n^2)$
Best case	$O(n)$
Average case	$O(n^2)$

Programming Assignment

Implement Insertion Sort

- Count the number of key comparisons and assignments for various inputs and plot the graph for both of them.
- For every input size n , run it with 10 different data points generated randomly.
- Compute the minimum, maximum and average of the number of key comparisons (and assignments) for each input size.
- Plot the graph for each case - best, worst and average number of comparisons (and assignments) .
- n varies from 10 to 100 in steps of 5.