# Algorithms

FOURTH EDITION

ROBERT SEDGEWICK | KEVIN WAYNE

http://algs4.cs.princeton.edu

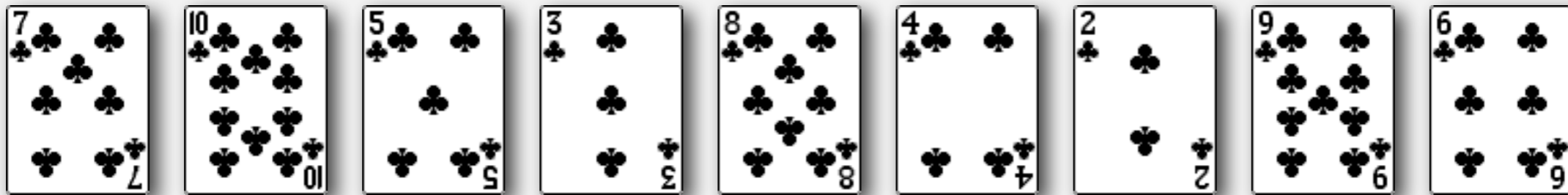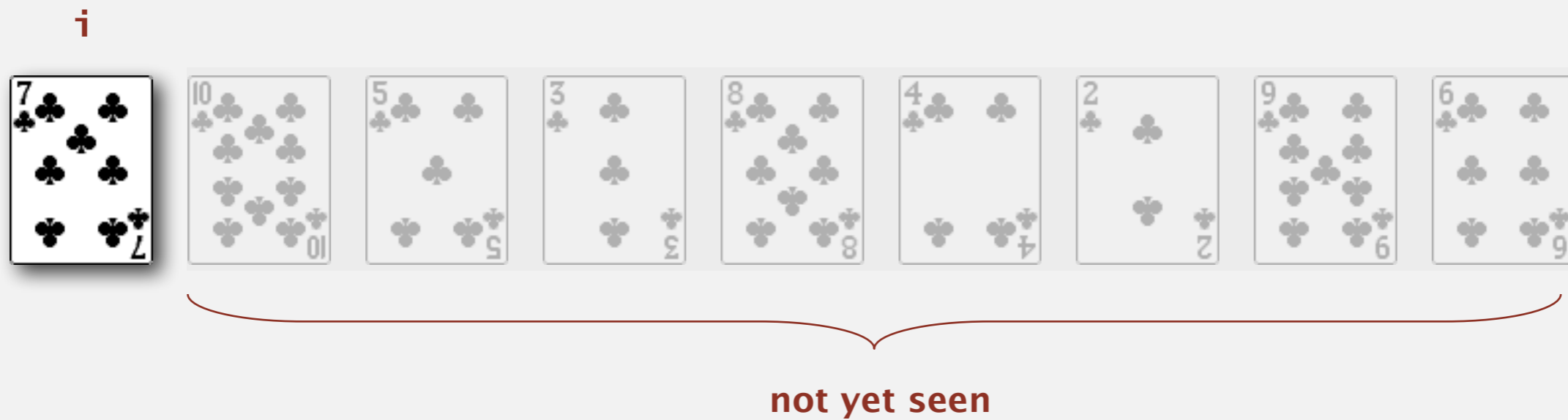# 2.1 INSERTION SORT DEMO

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



**not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

**j  i**



**in ascending order**                    **not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

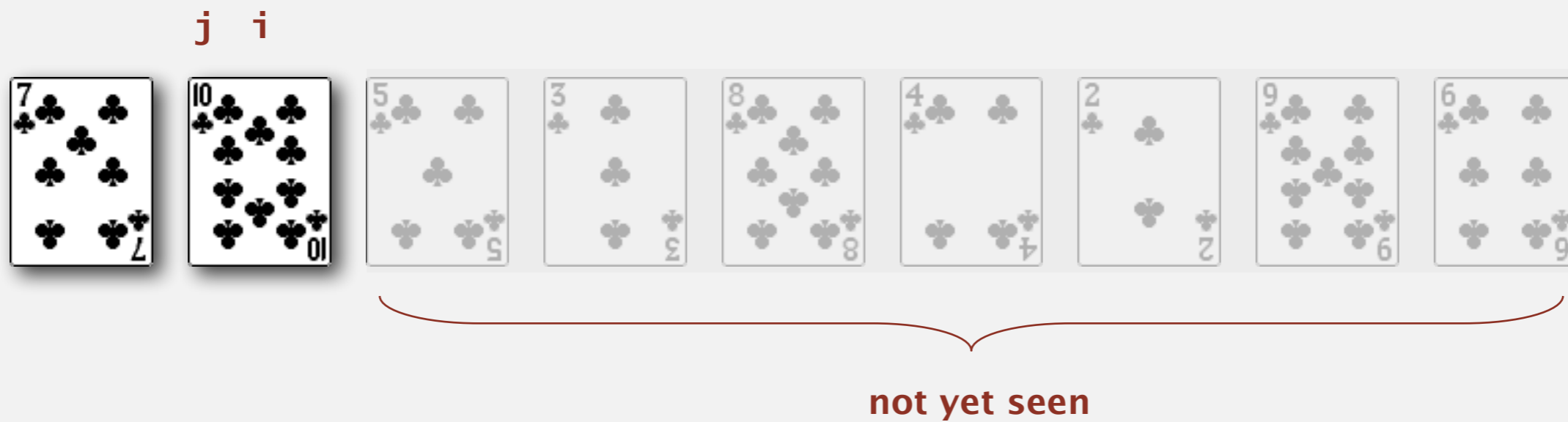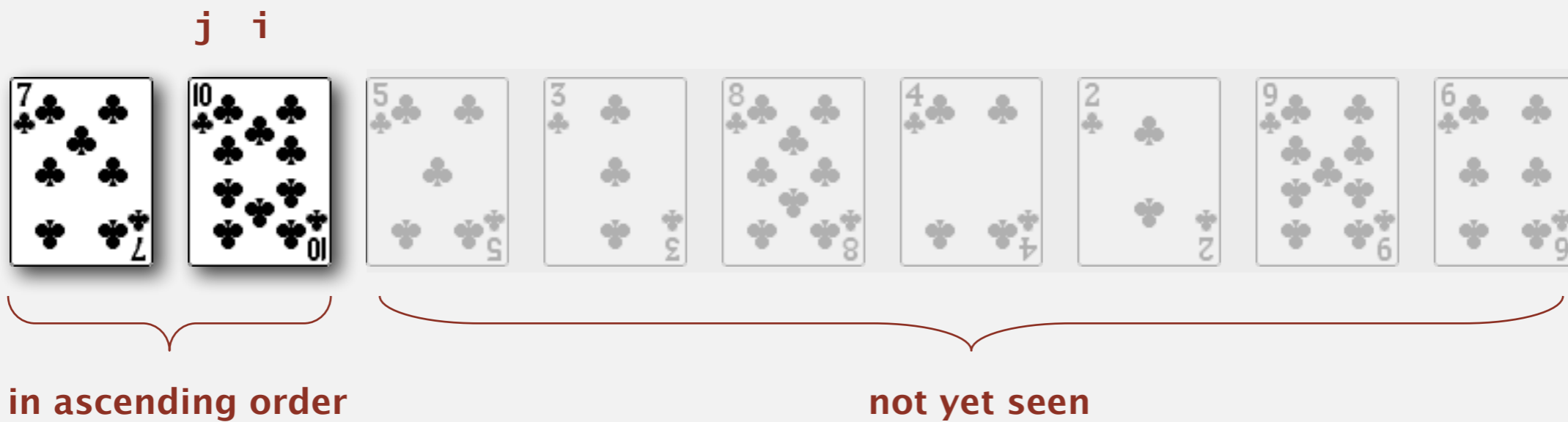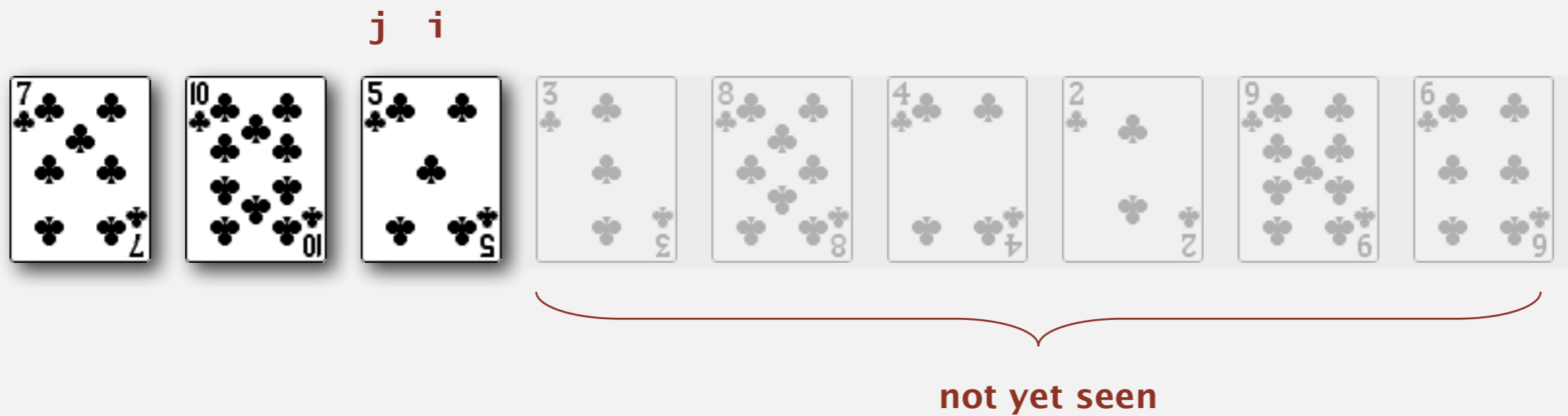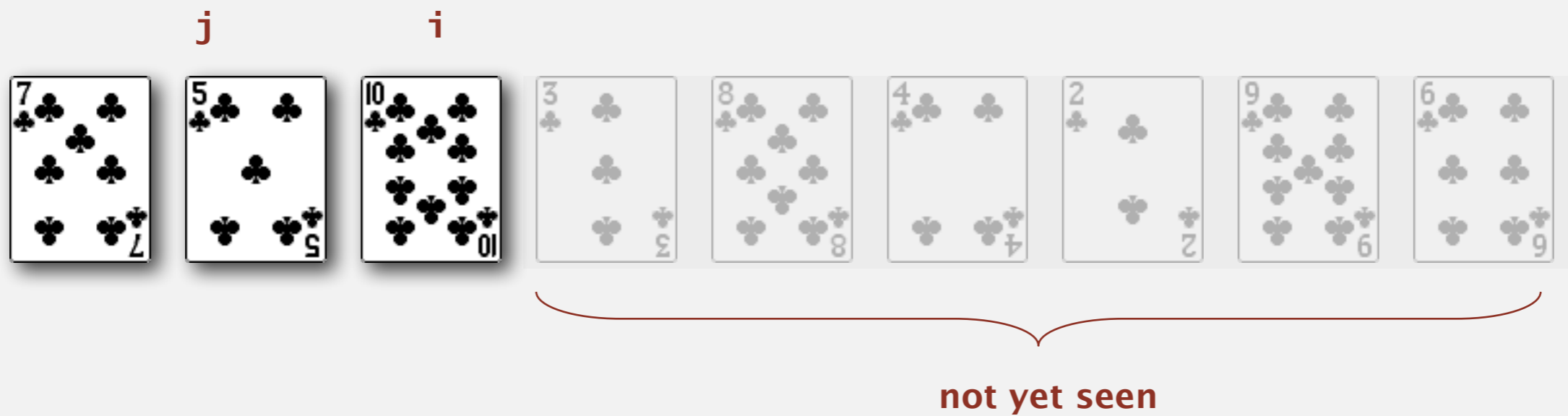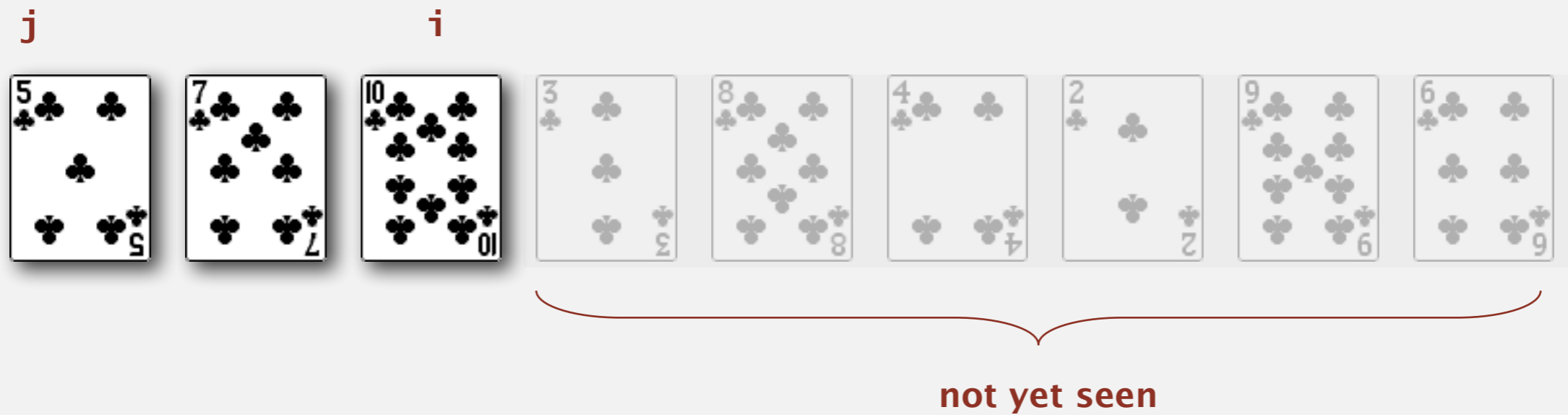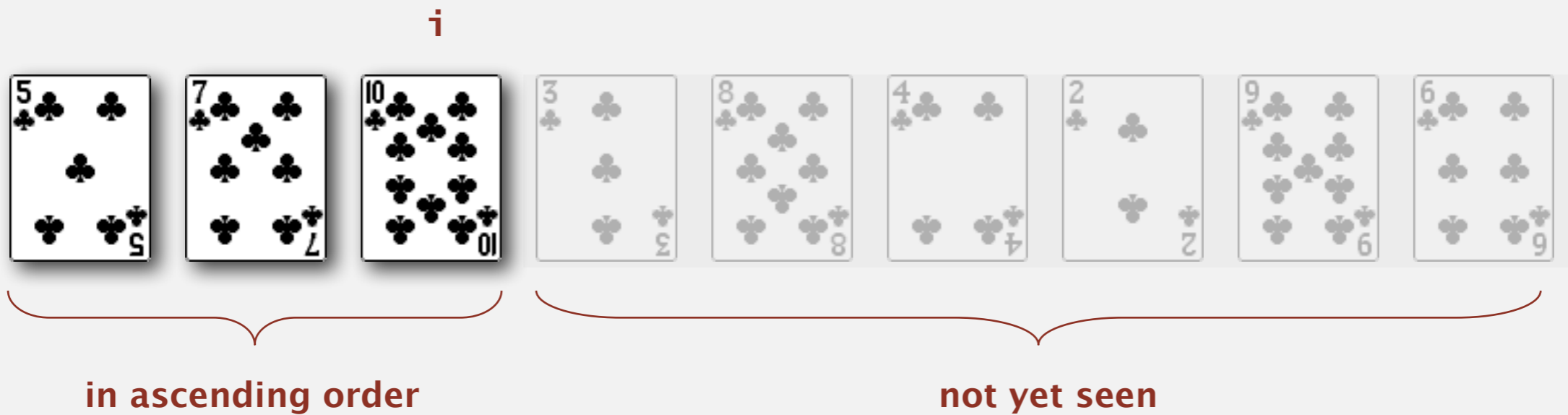- In iteration `i`, swap `a[i]` with each larger entry to its left.



**in ascending order**          **not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



**not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

i

in ascending order          not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

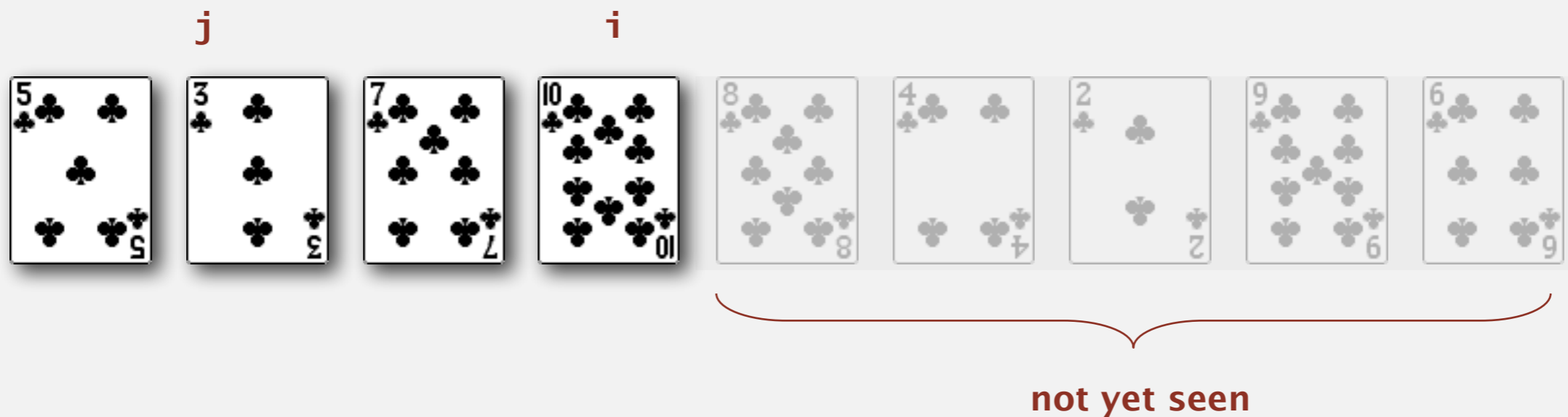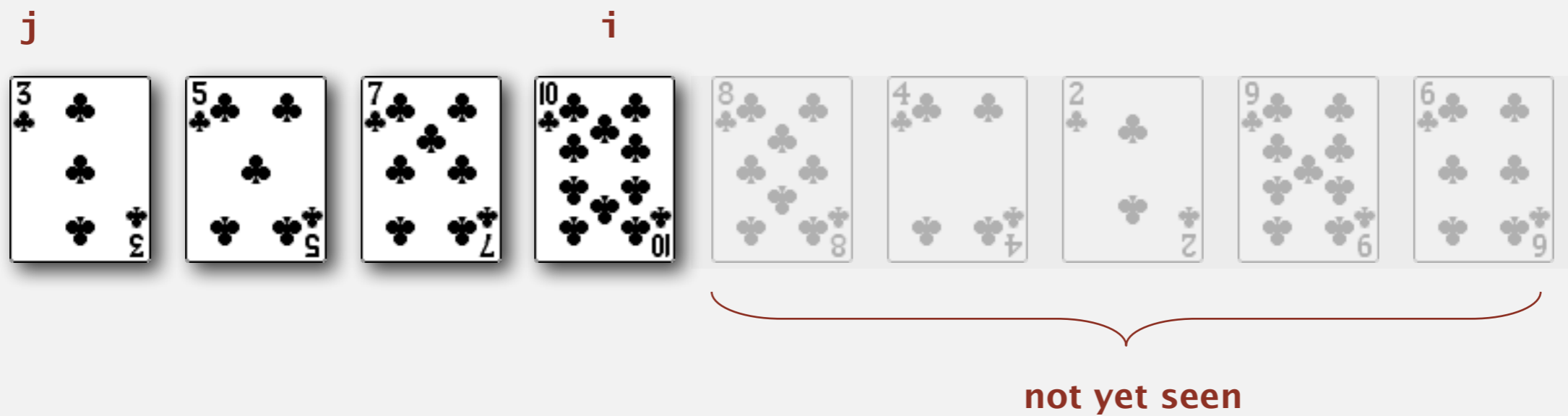- In iteration `i`, swap `a[i]` with each larger entry to its left.



j   i

not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

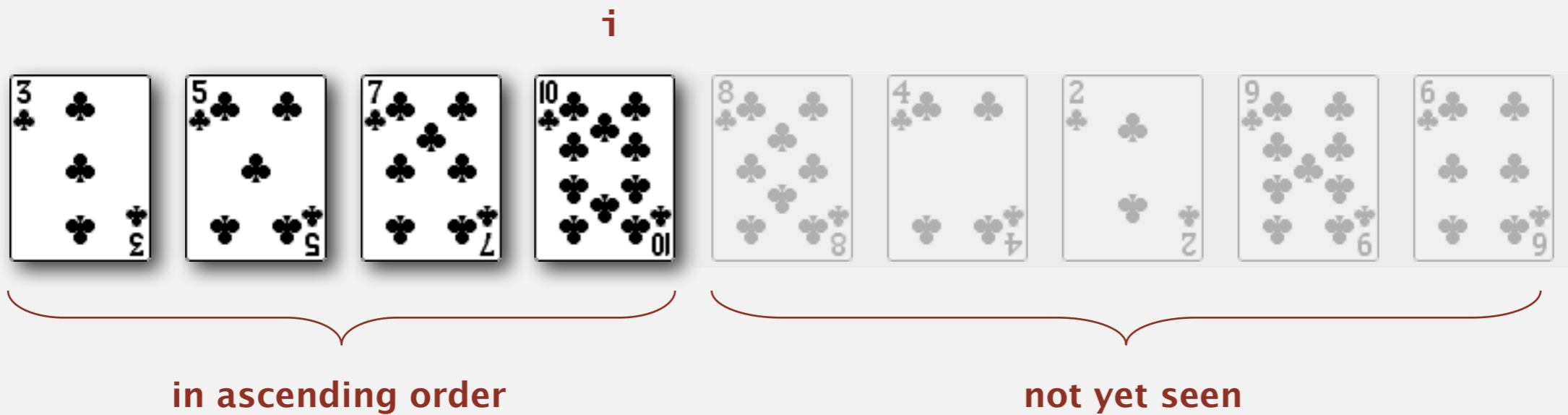- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.


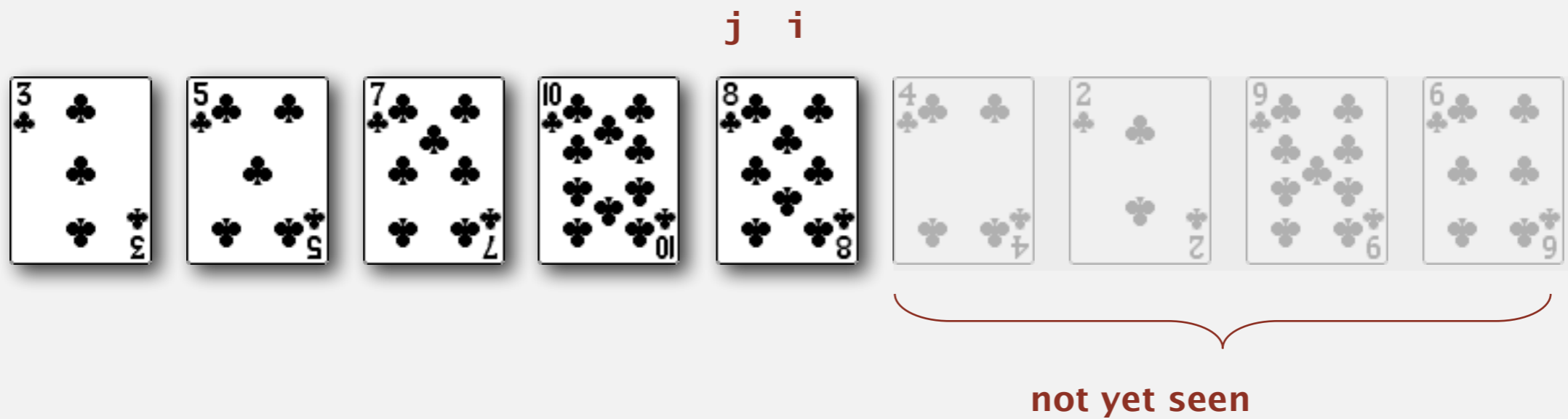
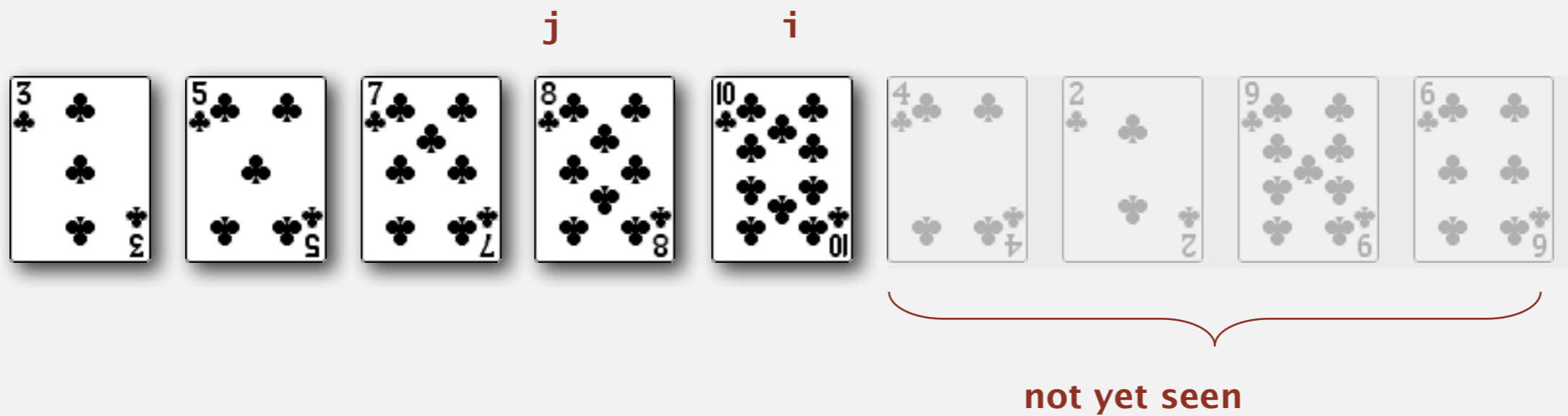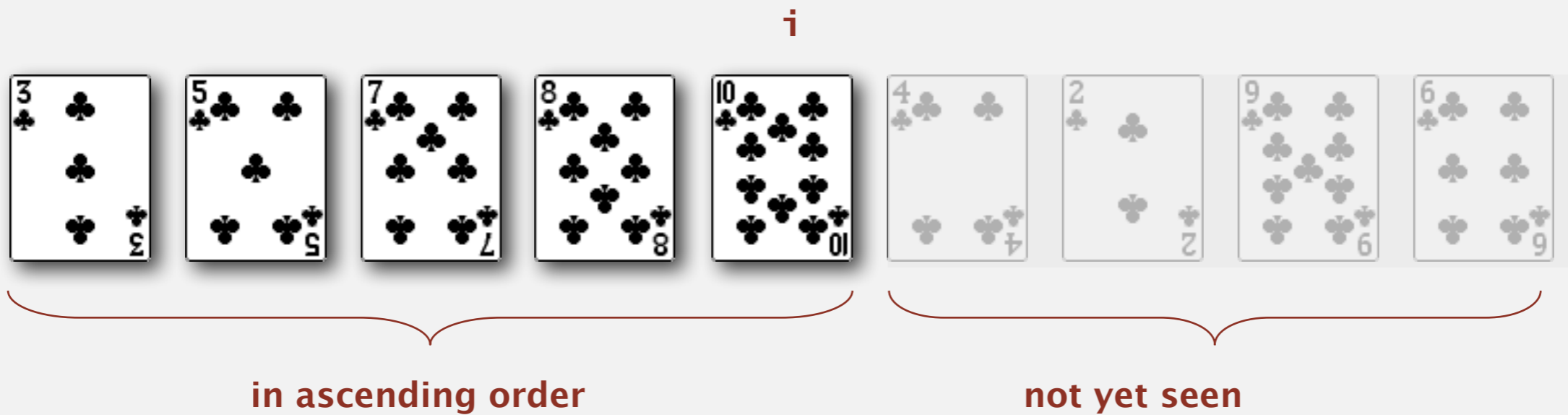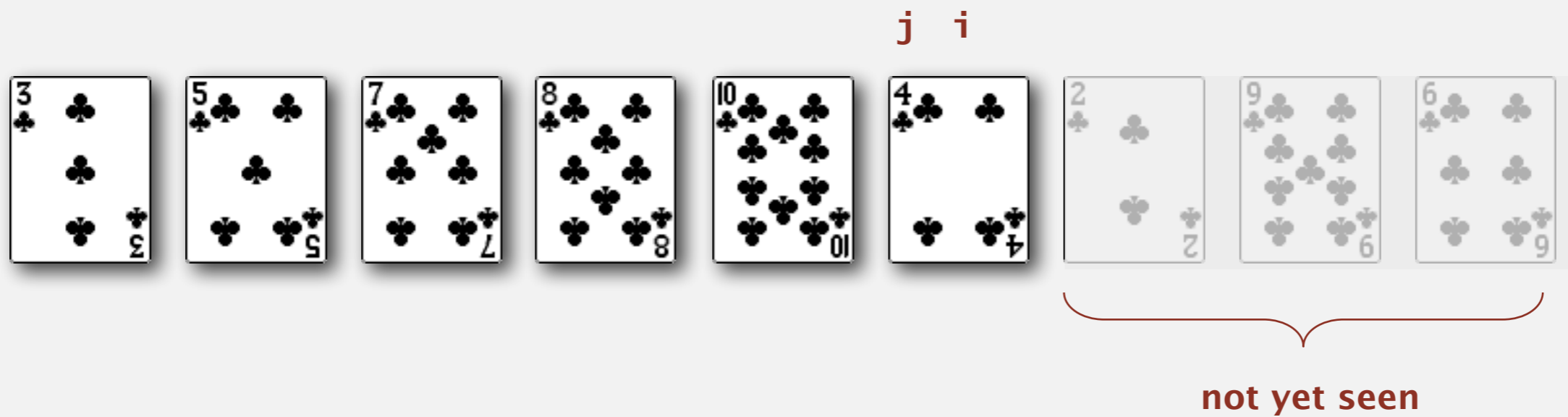in ascending order                     not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

j        i

not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



**i**

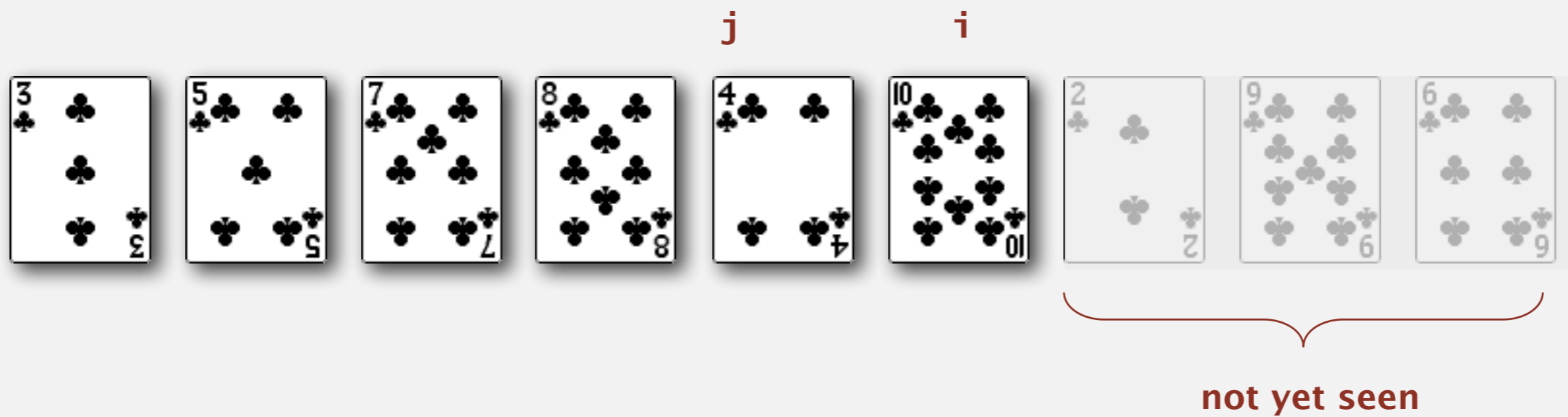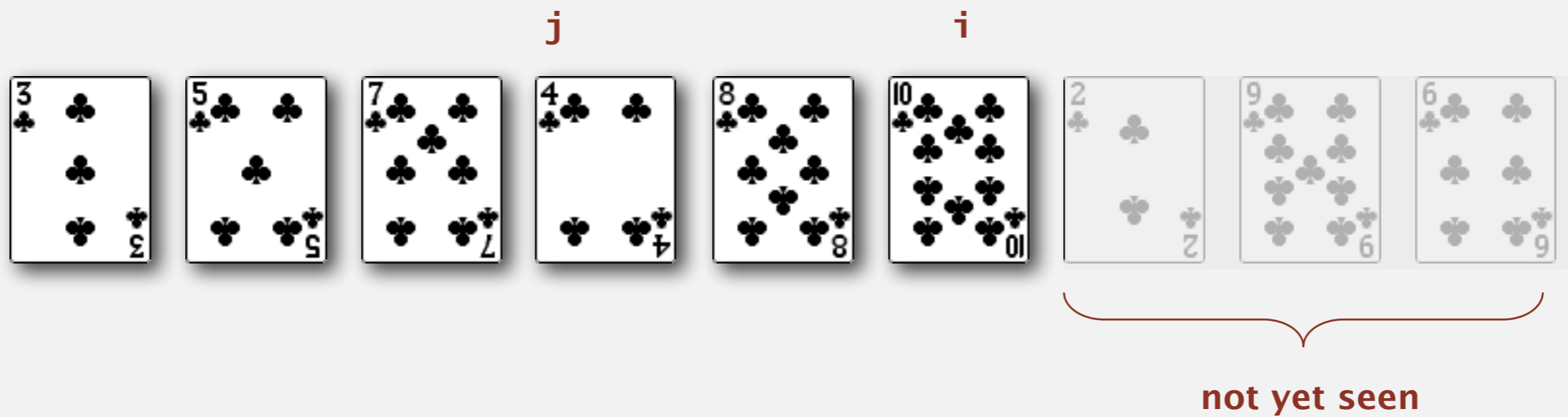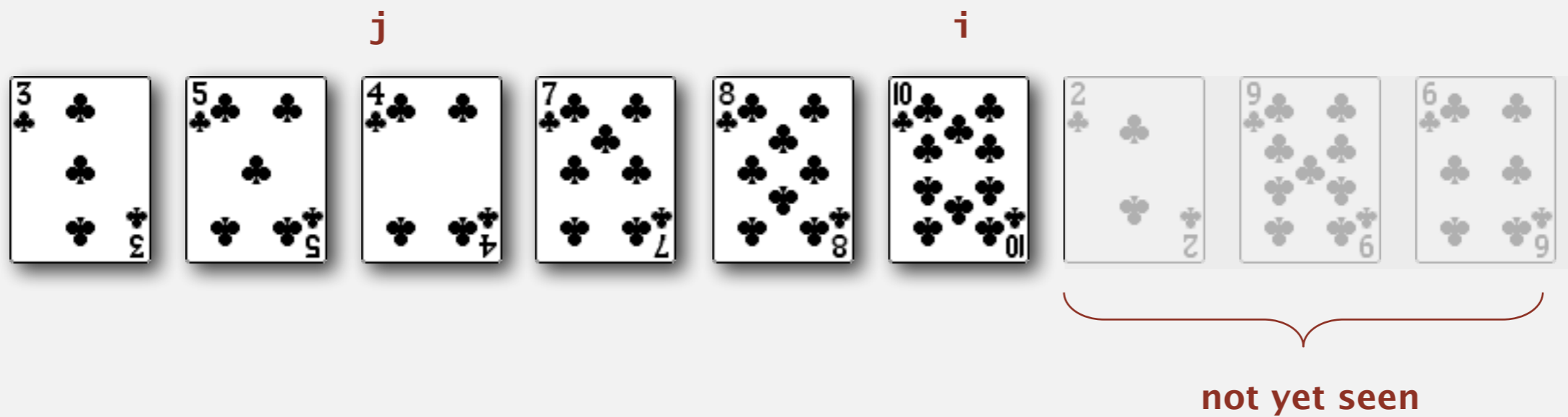**in ascending order**          **not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.
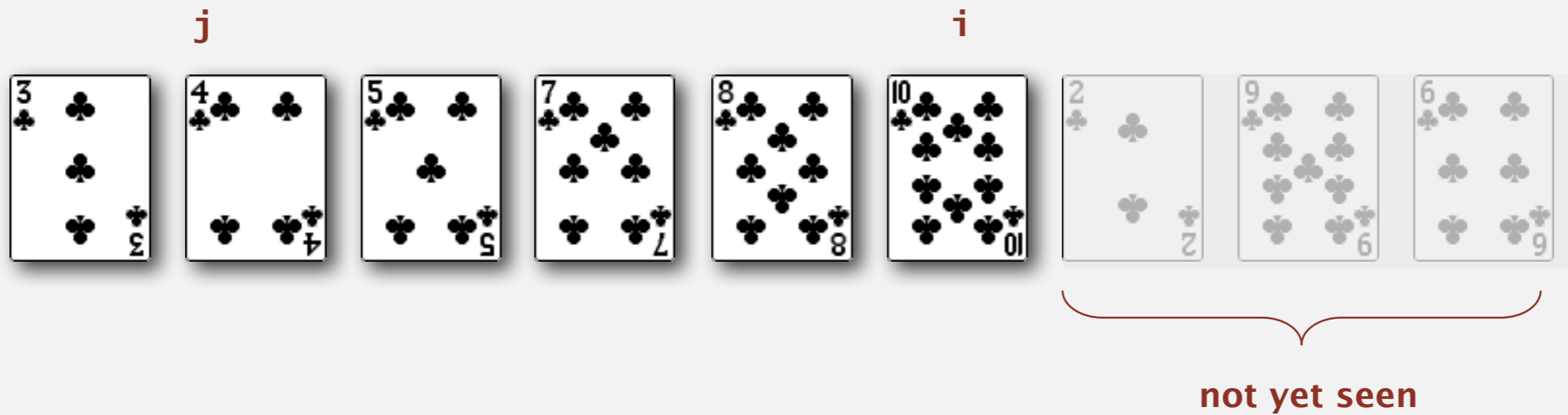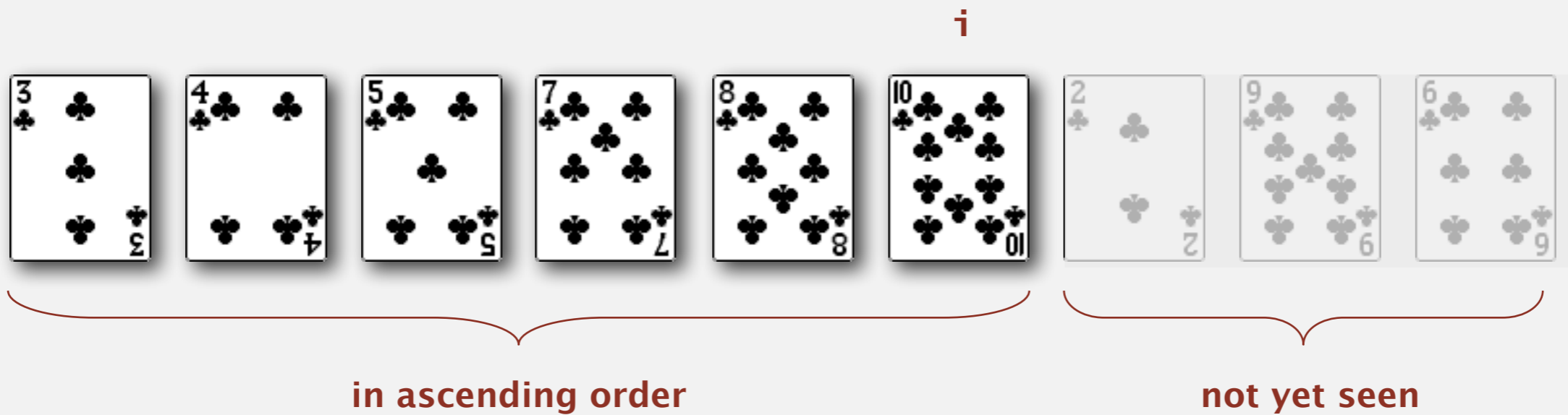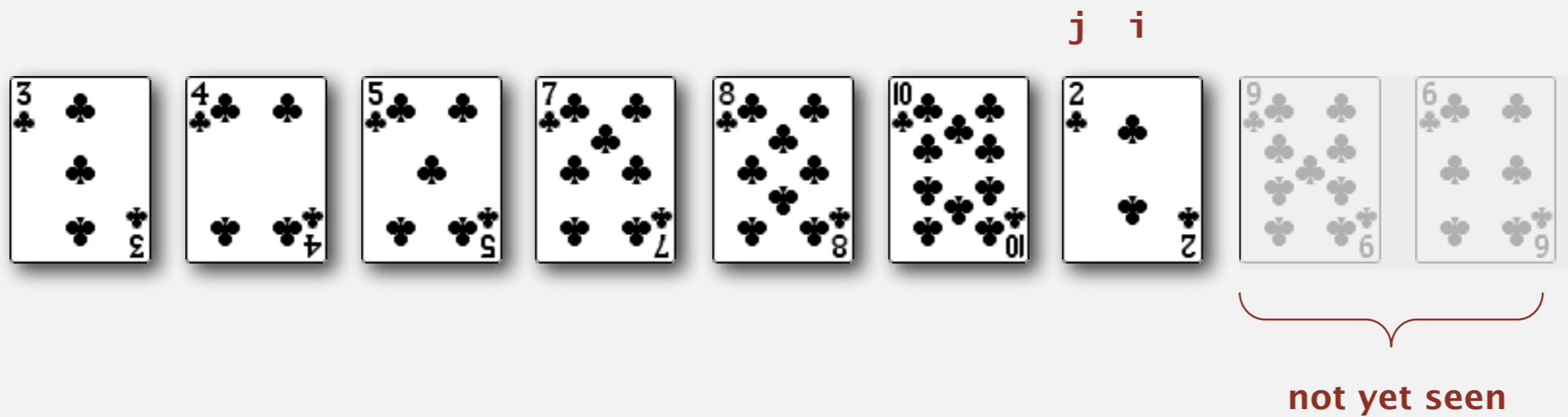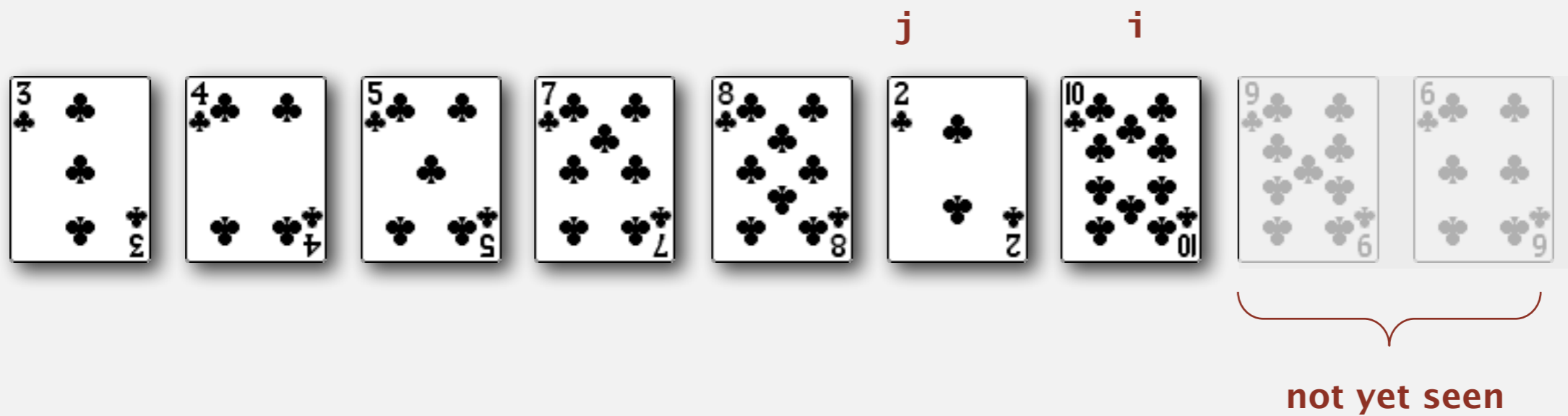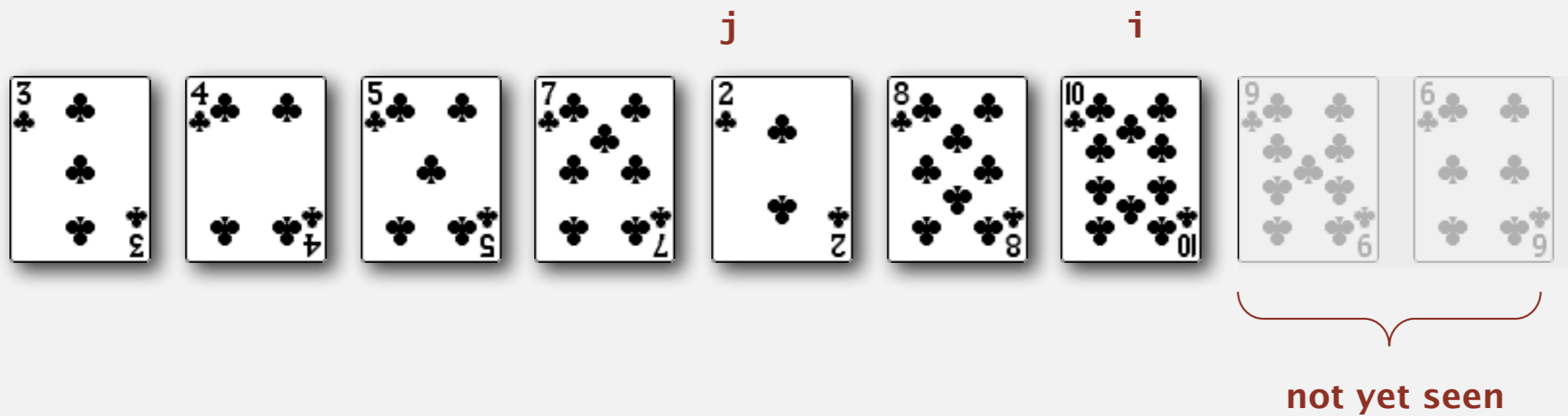


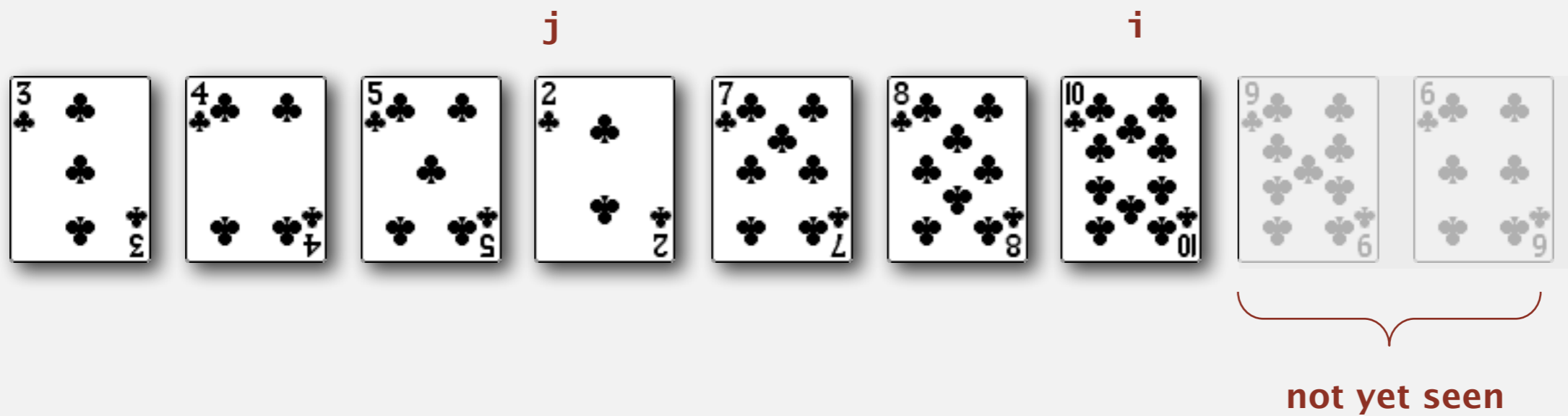not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

i

| 3♣ | 4♣ | 5♣ | 7♣ | 8♣ | 10♣ | 2♣ | 9♣ | 6♣ |

**in ascending order**  **not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

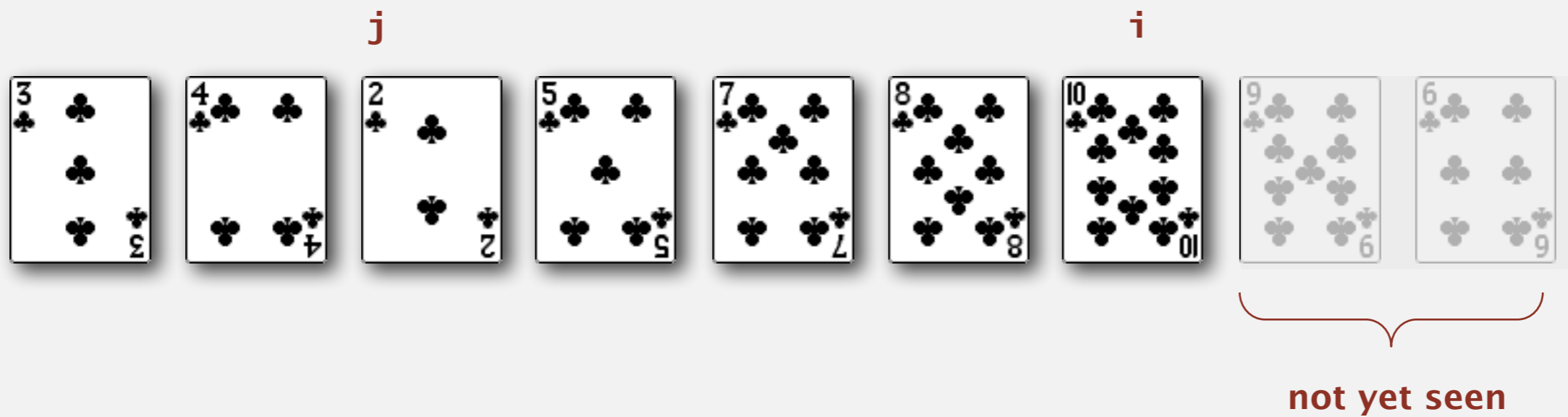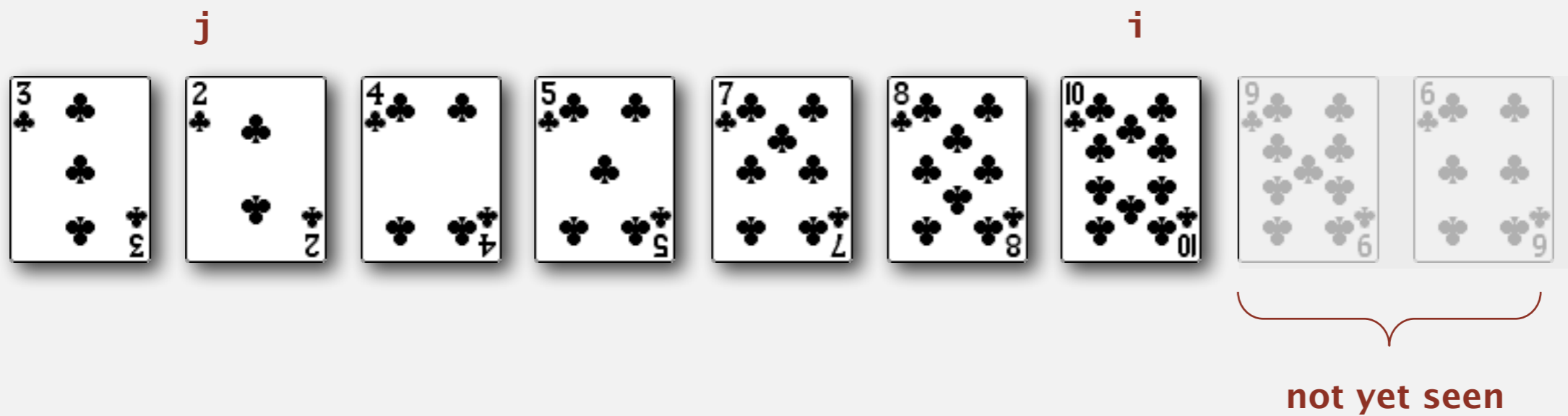- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

- In iteration `i`, swap `a[i]` with each larger entry to its left.



**not yet seen**

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

not yet seen

# Insertion sort demo

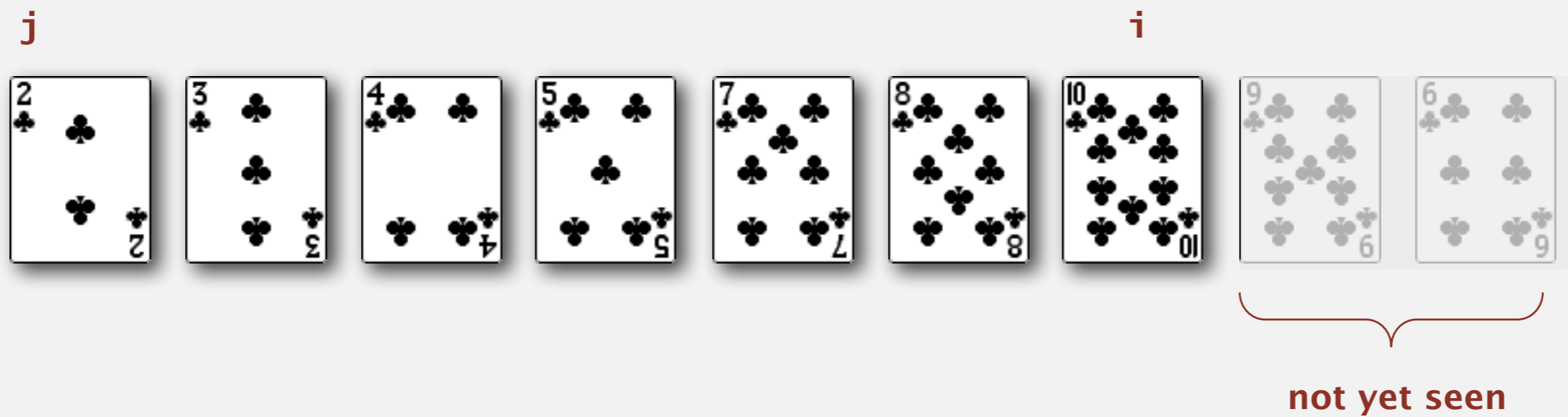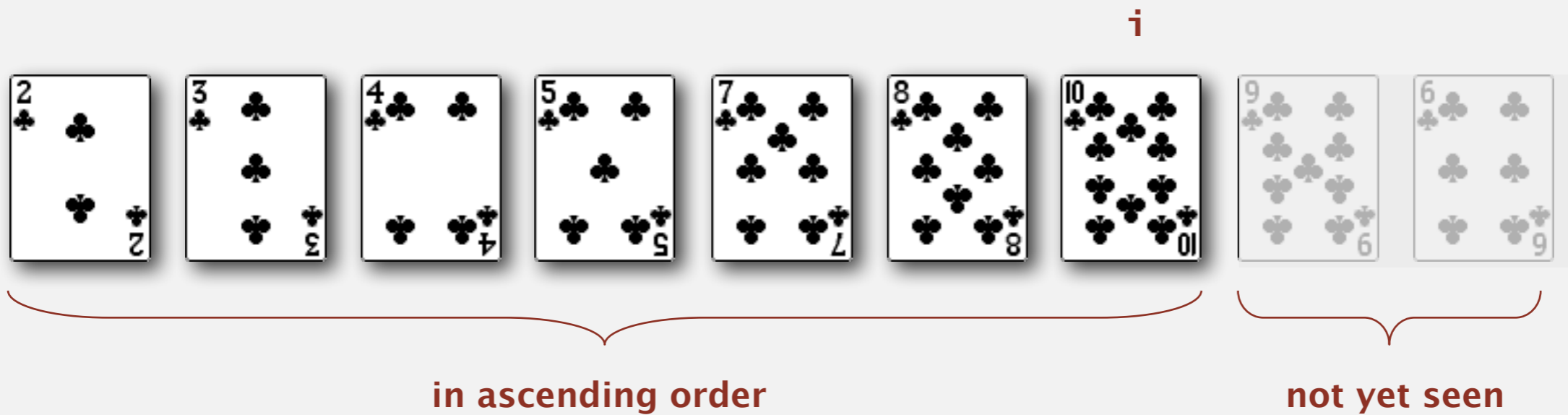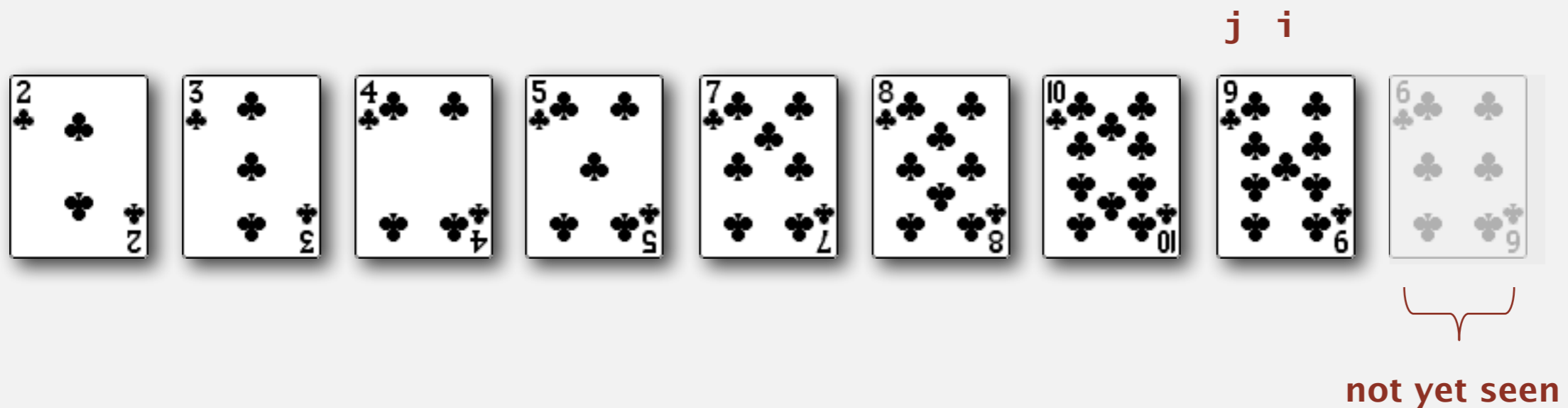- In iteration `i`, swap `a[i]` with each larger entry to its left.



**i**

**in ascending order**

**not yet seen**

- In iteration `i`, swap `a[i]` with each larger entry to its left.

j  i

not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



not yet seen

# Insertion sort demo

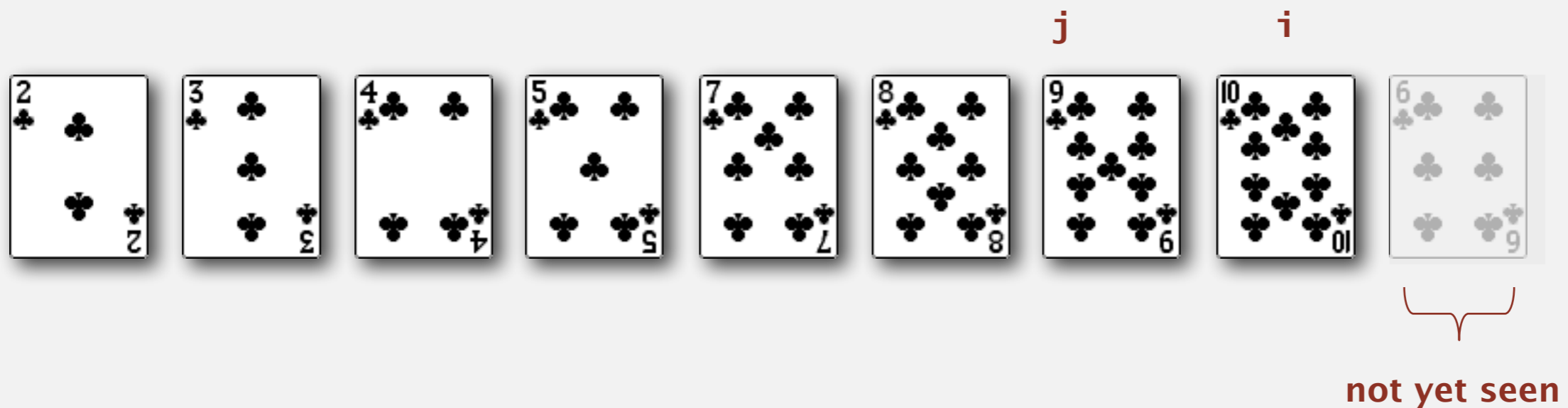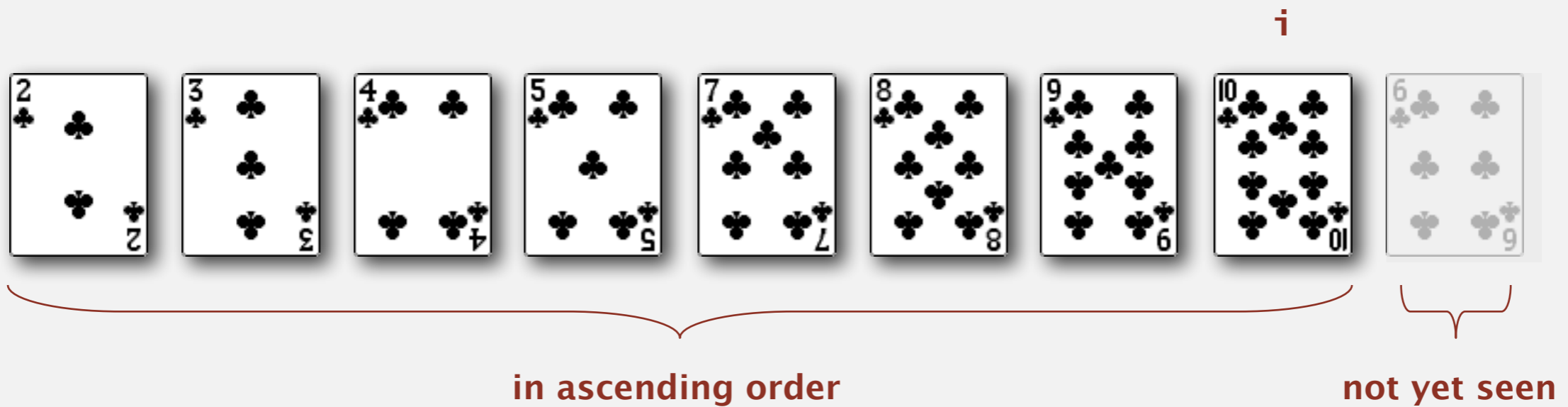- In iteration `i`, swap `a[i]` with each larger entry to its left.

i

in ascending order                                    not yet seen

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

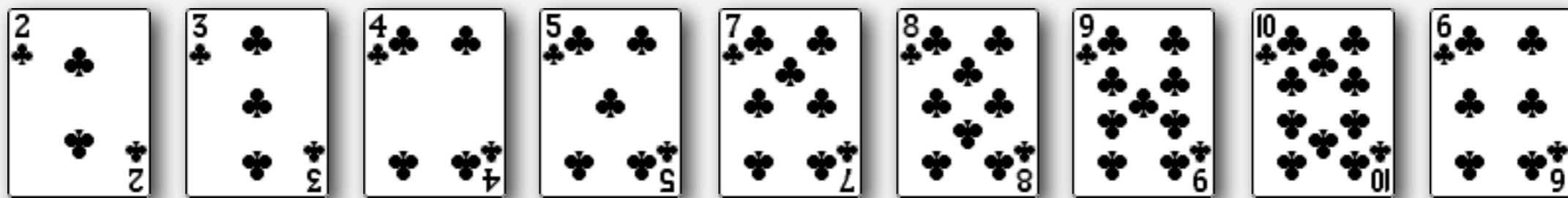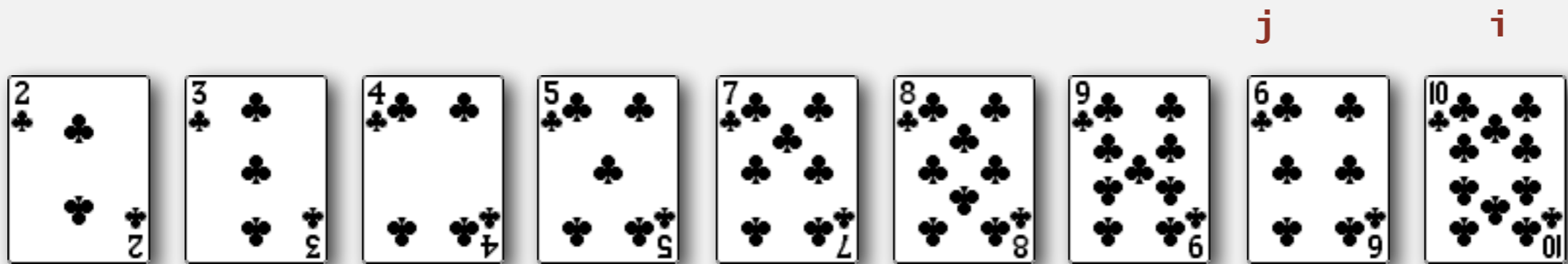- In iteration `i`, swap `a[i]` with each larger entry to its left.

- In iteration `i`, swap `a[i]` with each larger entry to its left.
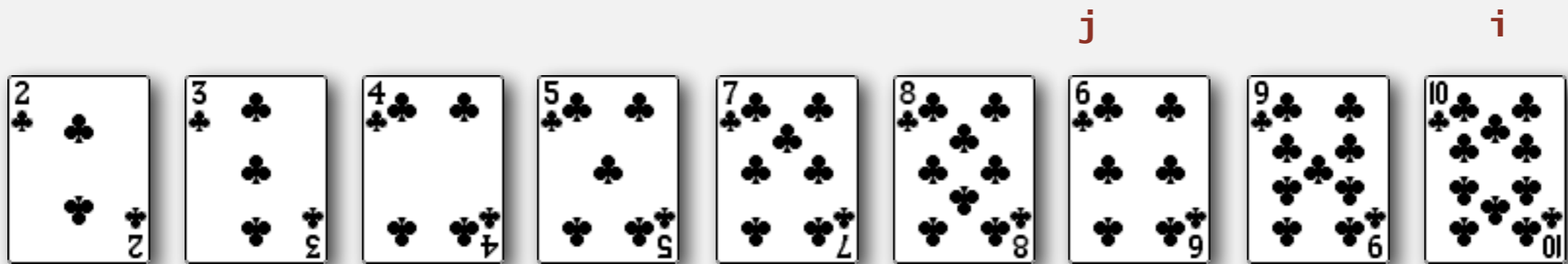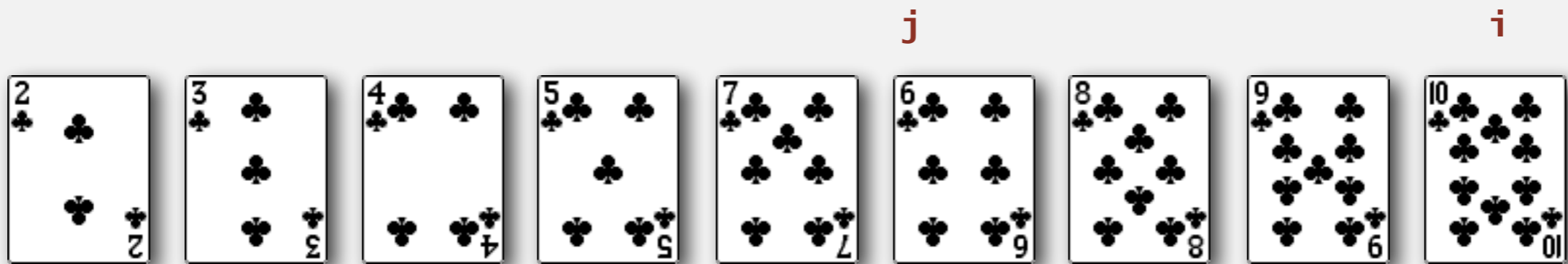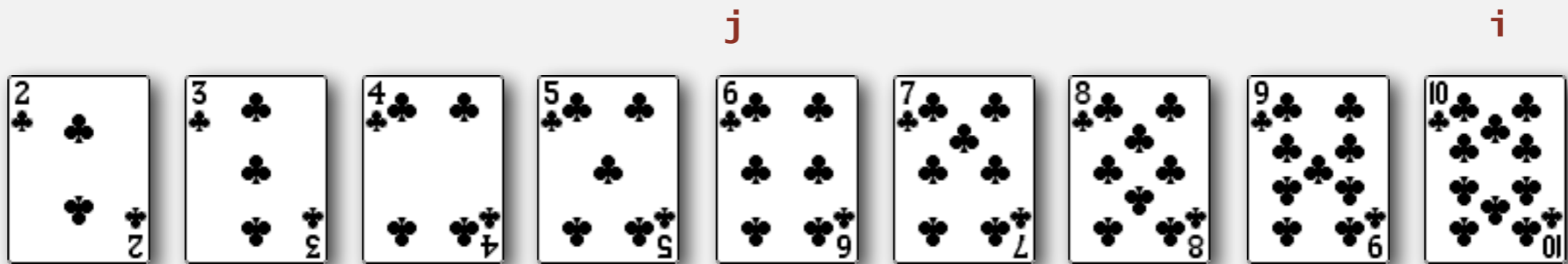
# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.

# Insertion sort demo

- In iteration `i`, swap `a[i]` with each larger entry to its left.



**sorted**