

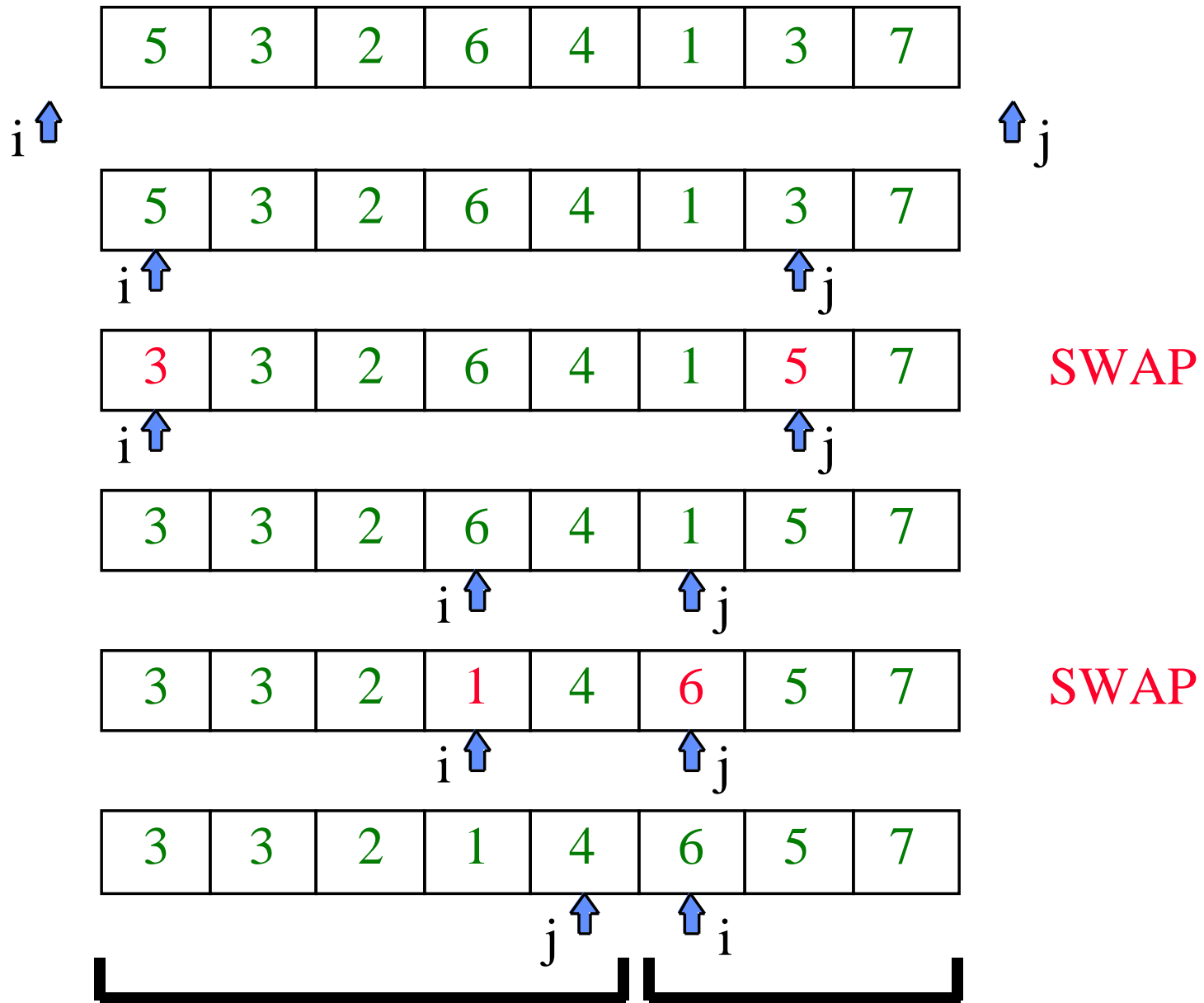
# Notes on QuickSort

```
procedure quicksort(A,p,r)
    VAR A: one-dimensional array
        p,r: natural numbers

procedure partition(A,p,r)
    VAR A: one-dimensional array
        p,r: natural numbers
```

```
procedure quicksort(A,p,r)
  if p<r then q := partition(A,p,r)
    quicksort(A,p,q)
    quicksort(A,q+1,r)
```

```
procedure partition(A,p,r)
  X := A[p]
  i := p-1
  j := r+1
  while true
    begin repeat j := j-1
      until A[j] <= X
    repeat i := i+1
      until A[i] >= X
    if i<j then swap(A,i,j)
      else return j
    end
  end
```



p r

```
procedure quicksort(A,p,r)
```

```
  if p<r then q := partition(A,p,r)
```

```
    quicksort(A,p,q)
```

```
    quicksort(A,q+1,r)
```

p < r

```
procedure partition(A,p,r)
```

```
  X := A[p]
```

```
  i := p-1
```

```
  j := r+1
```

```
  while true
```

```
    begin repeat j := j-1
```

```
      until A[j] <= X
```

```
    repeat i := i+1
```

```
      until A[i] >= X
```

```
    if i < j then swap(A,i,j)
```

```
      else return j
```

```
  end
```

```
i < j
```

```
u: [u {A[p], ... ,A[i]} u X]
```

```
v: [v {A[j], ... ,A[r]} v X]
```

procedure partition(A,p,r)

X := A[p]

i := p-1

j := r+1

while true

begin repeat j := j-1

until A[j] <= X

repeat i := i+1

until A[i] >= X

if i < j then swap(A,i,j)

else return j

end

i < j

u: [u {A[p], ... ,A[i]} u X]

v: [v {A[j], ... ,A[r]} v X]

i j

v: [v {A[j+1], ... ,A[r]} v X]

i j+1

u: [u {A[p], ... ,A[i-1]} u X]

p r

```
procedure quicksort(A,p,r)
```

```
  if p<r then q := partition(A,p,r)
```

p < r

```
    quicksort(A,p,q)
```

```
    quicksort(A,q+1,r)
```

u, v:

[u {A[p], ..., A[q]} ^

v {A[q+1], ..., A[r]}

u v]

Prove correct by induction?

procedure partition(A,p,r)

X := A[p]

i := p-1

j := r+1

while true

i < j

u: [u {A[p], ..., A[i]} u X]

v: [v {A[j], ..., A[r]} v X]

begin repeat j := j-1

p j r until A[j] <= X

repeat i := i+1

p i r until A[i] >= X

if i < j then swap(A, i, j)

else return j p j < r

end

i j

v: [v {A[j+1], ..., A[r]} v X]

i j+1

u: [u {A[p], ..., A[i-1]} u X]

p r

```
procedure quicksort(A,p,r)
```

```
  if p<r then q := partition(A,p,r)
```

p < r

```
    quicksort(A,p,q)
```

```
    quicksort(A,q+1,r)
```

u, v:

[u {A[p], ..., A[q]} ^

v {A[q+1], ..., A[r]}

u v]

p q < r

Prove correct by induction!

QED.