



UNIVERSIDADE DA CORUÑA



faculdade de
informática
da coruña

Small Group Tutorial 1: Insertion and Selection Sort

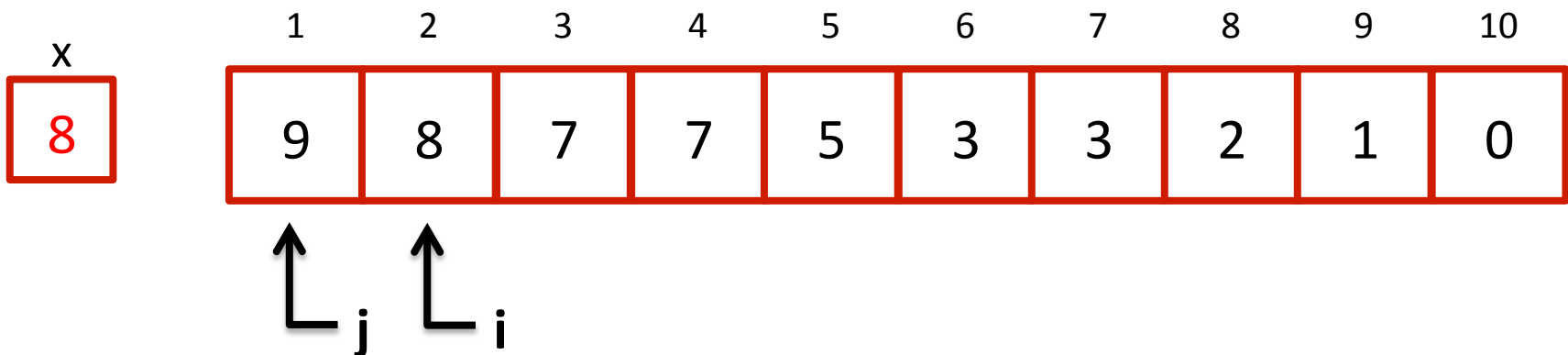
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```

1	2	3	4	5	6	7	8	9	10
9	8	7	7	5	3	3	2	1	0

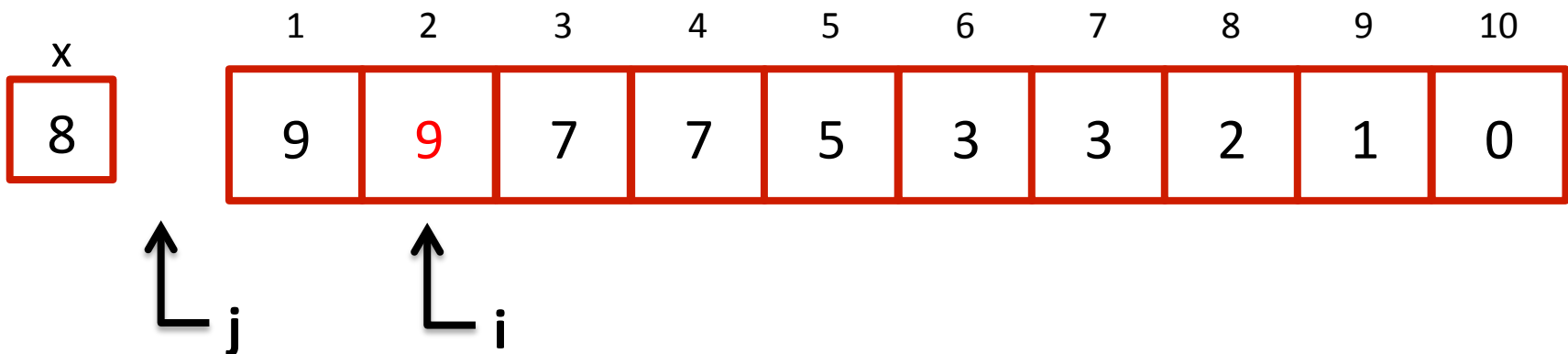
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



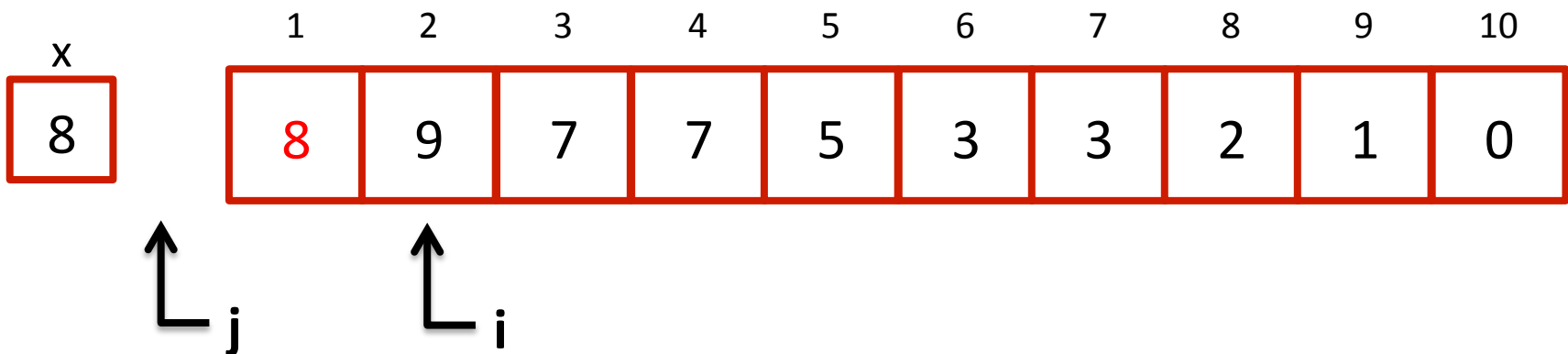
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



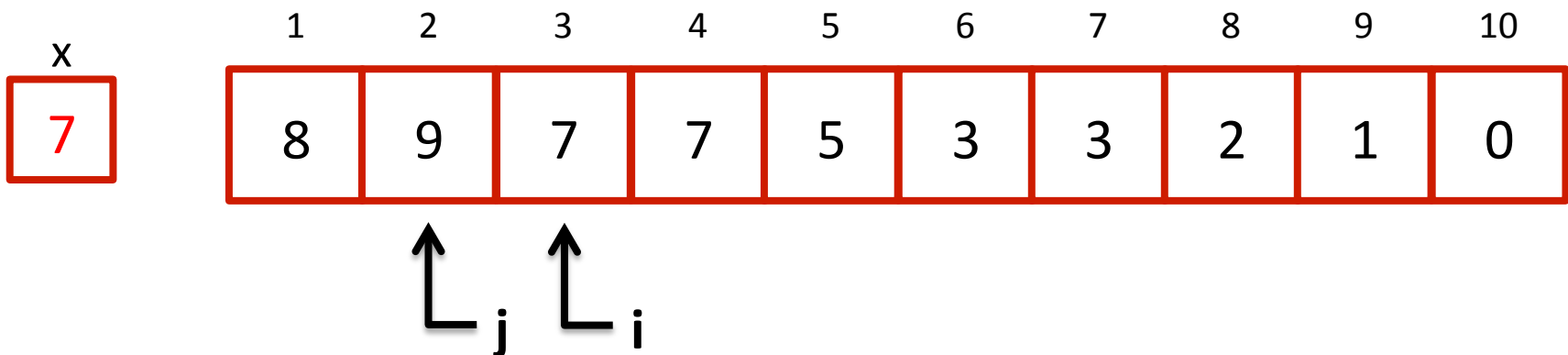
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



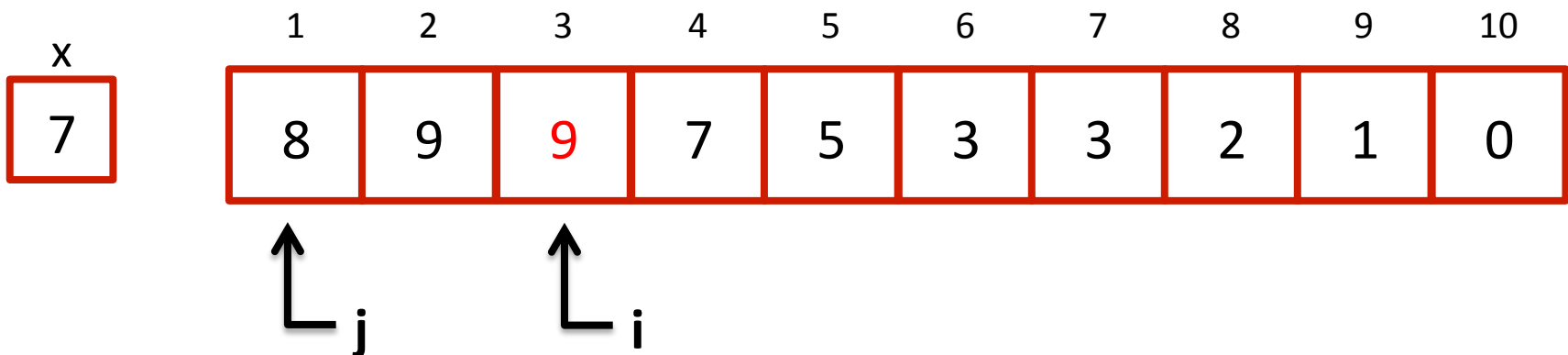
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



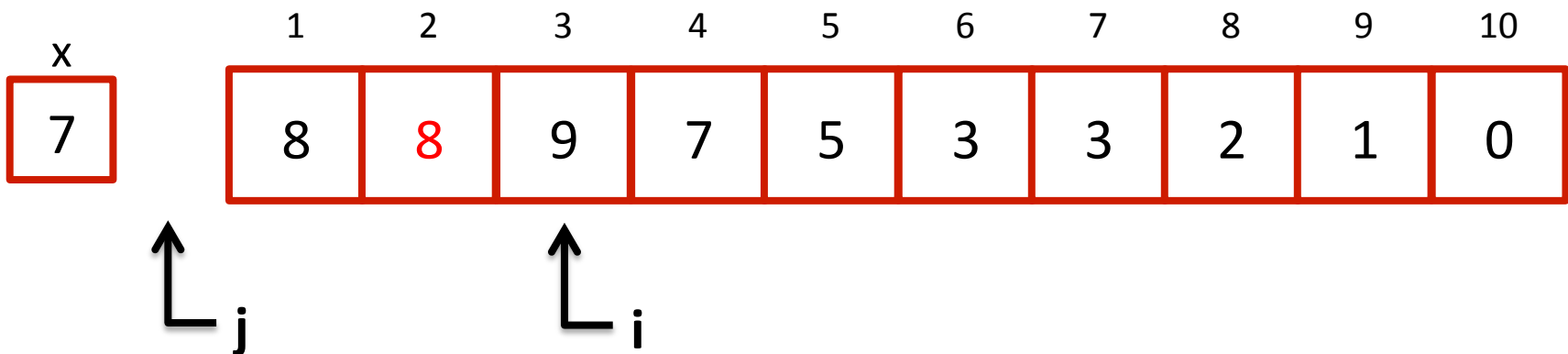
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



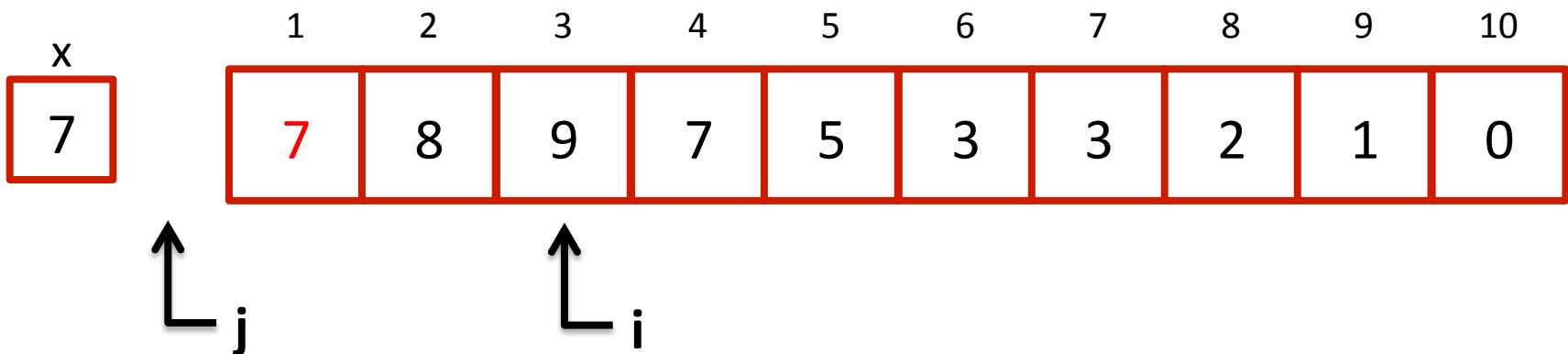
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



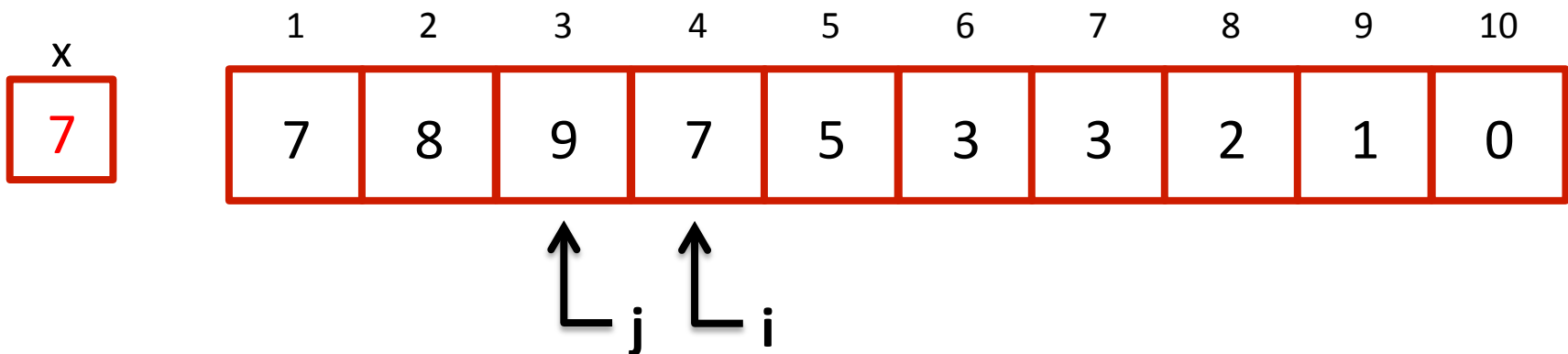
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



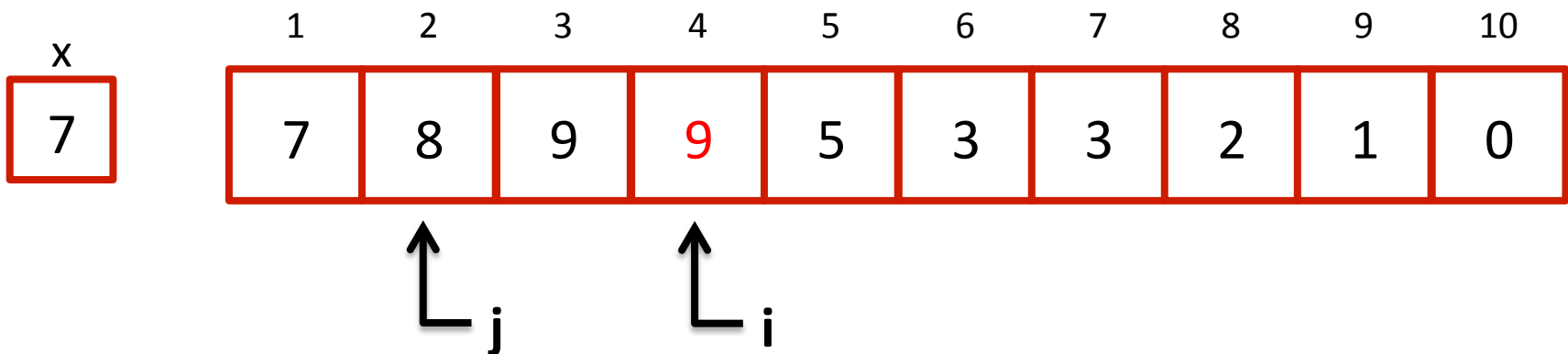
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



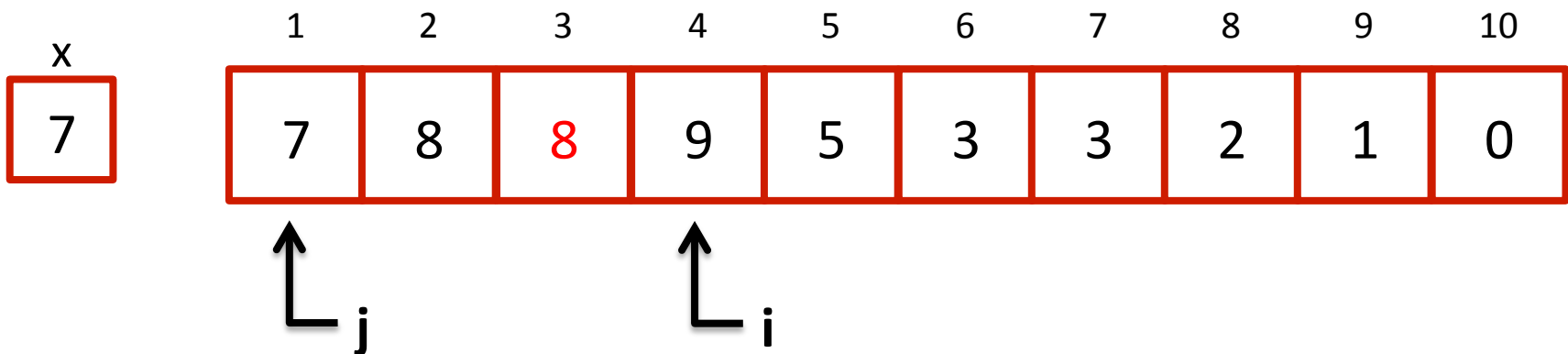
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



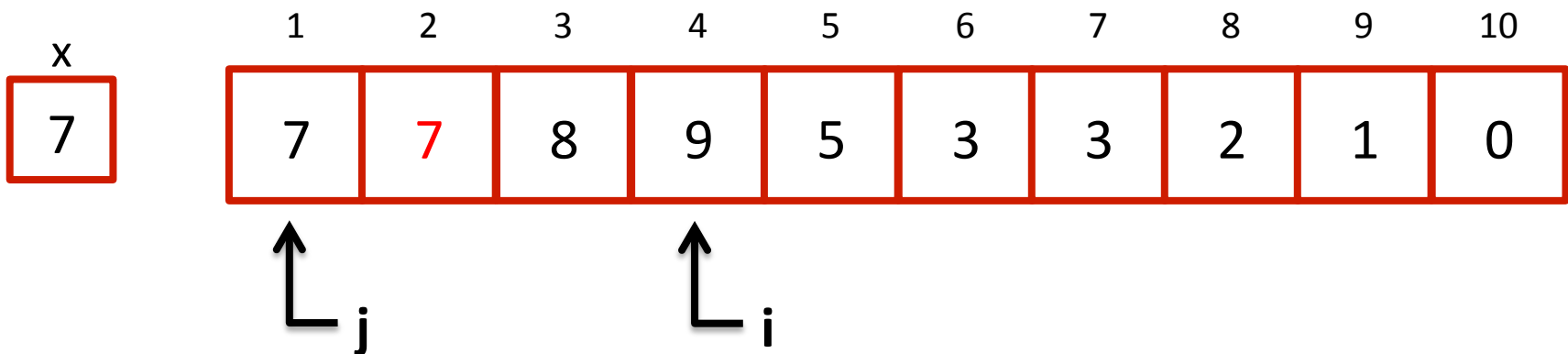
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



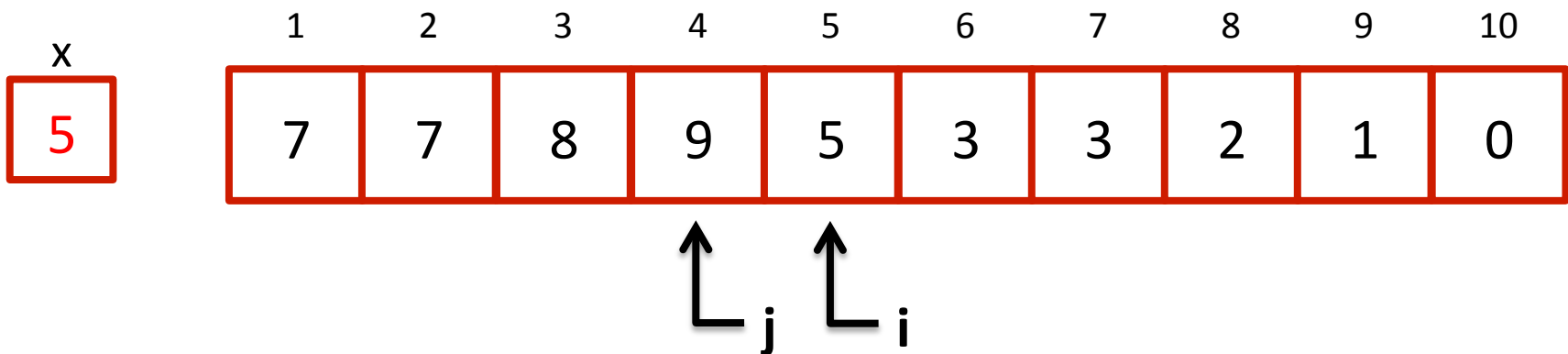
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



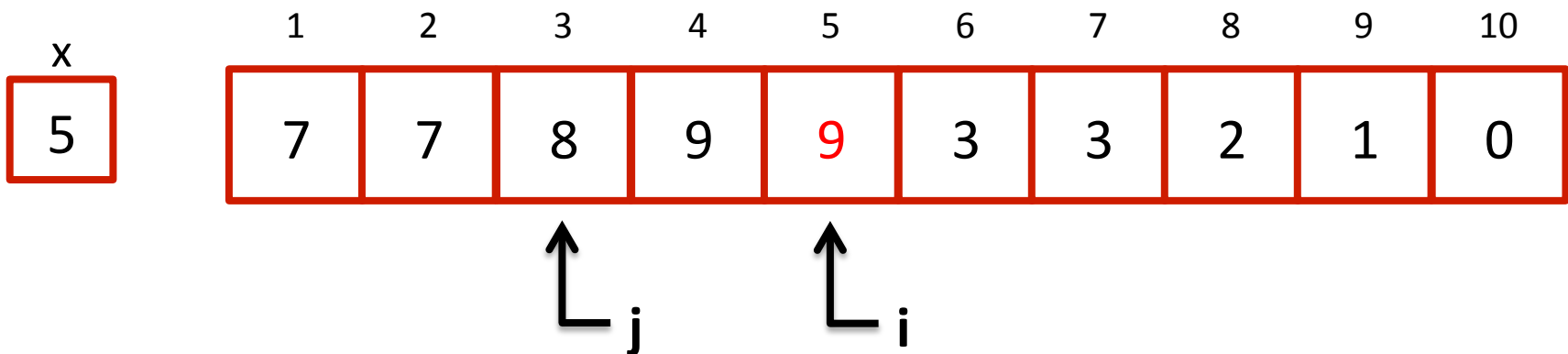
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



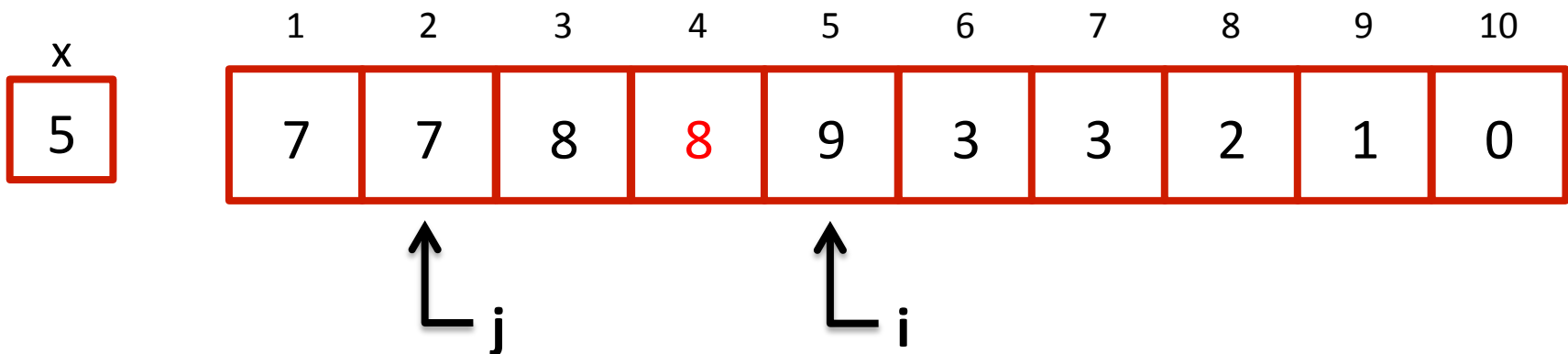
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



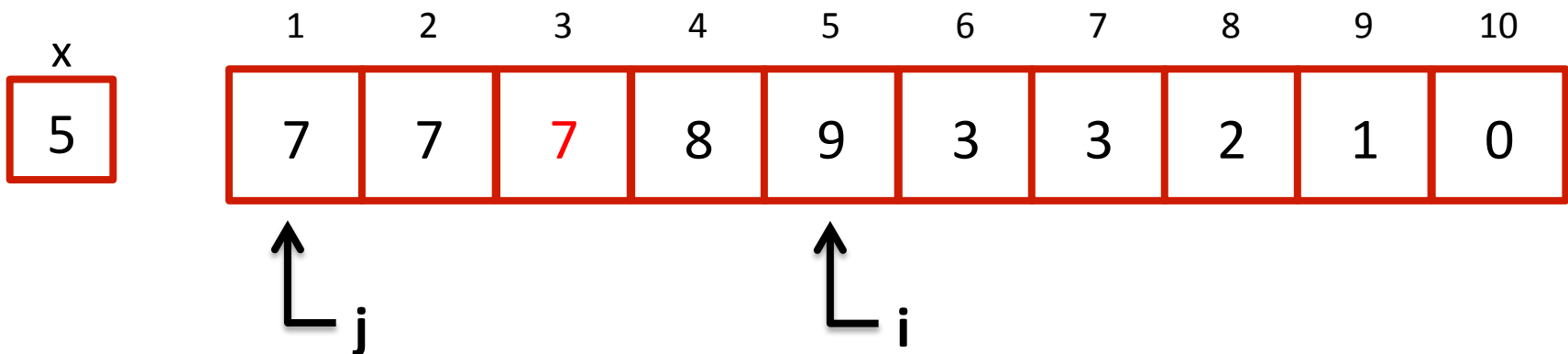
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



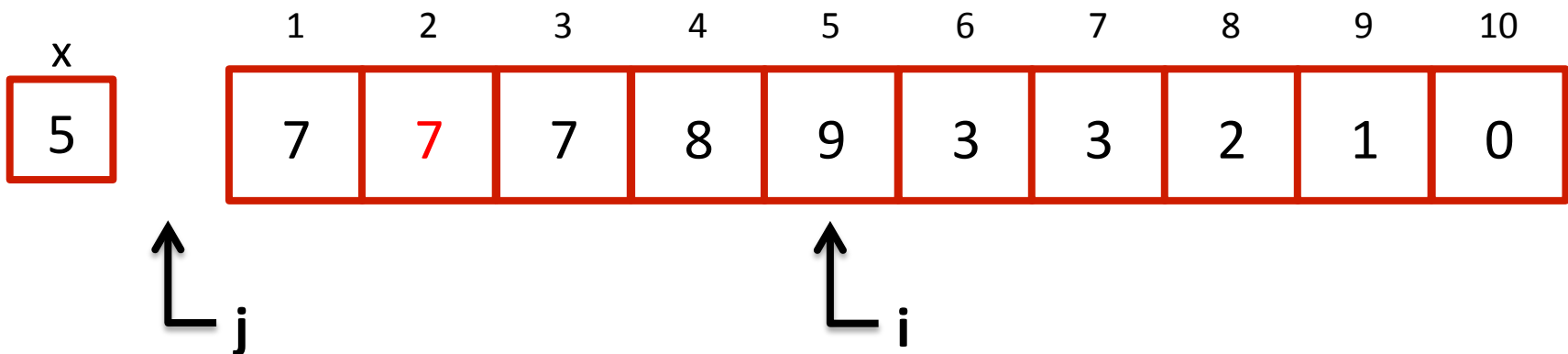
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



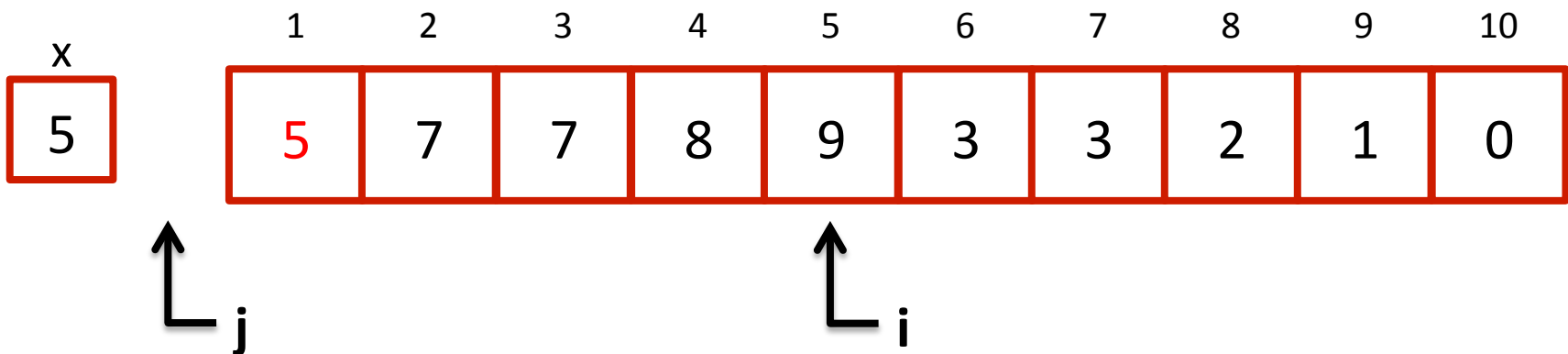
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



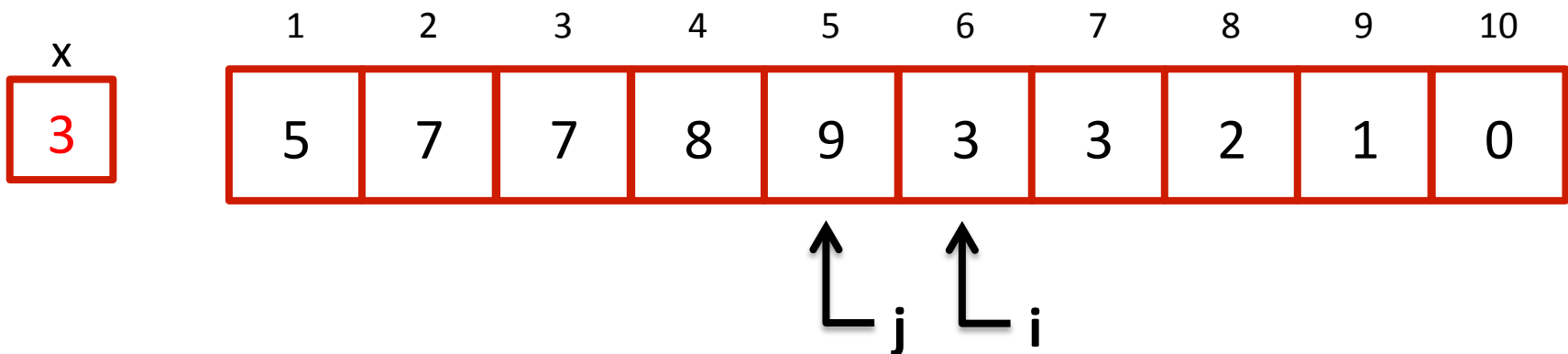
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



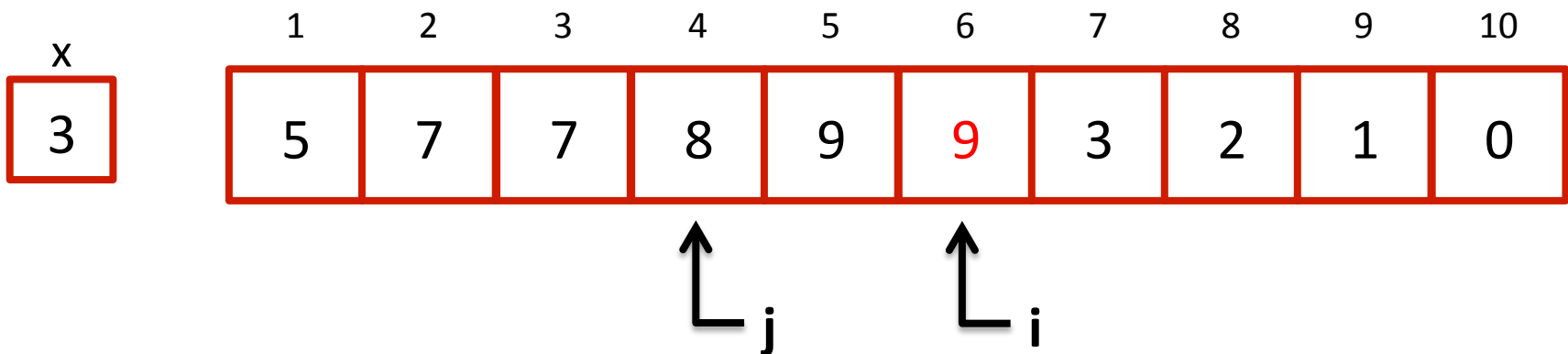
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



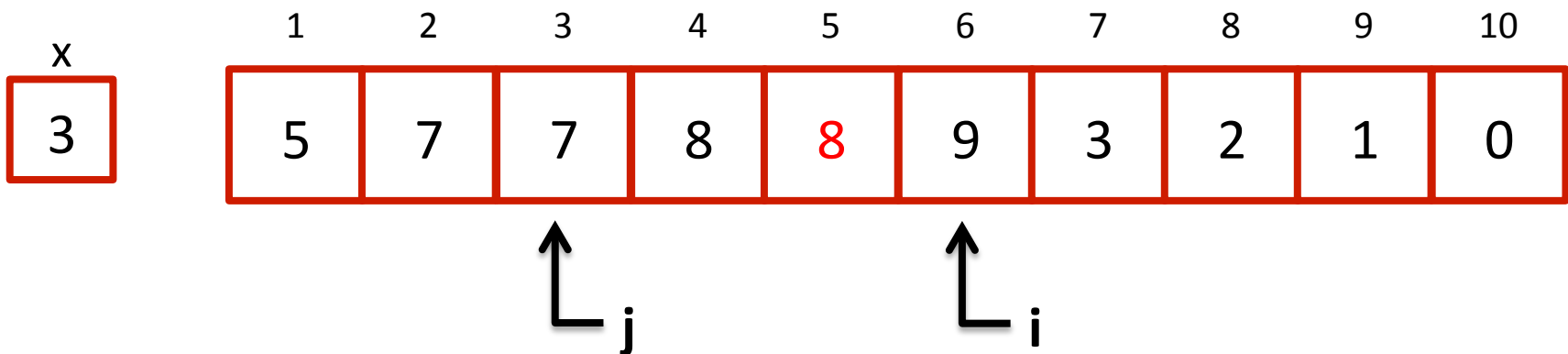
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



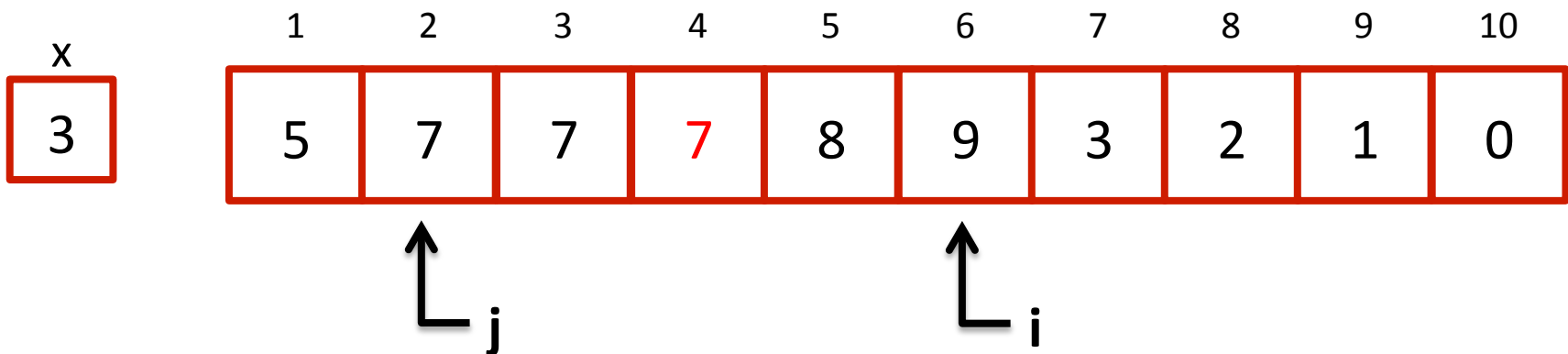
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



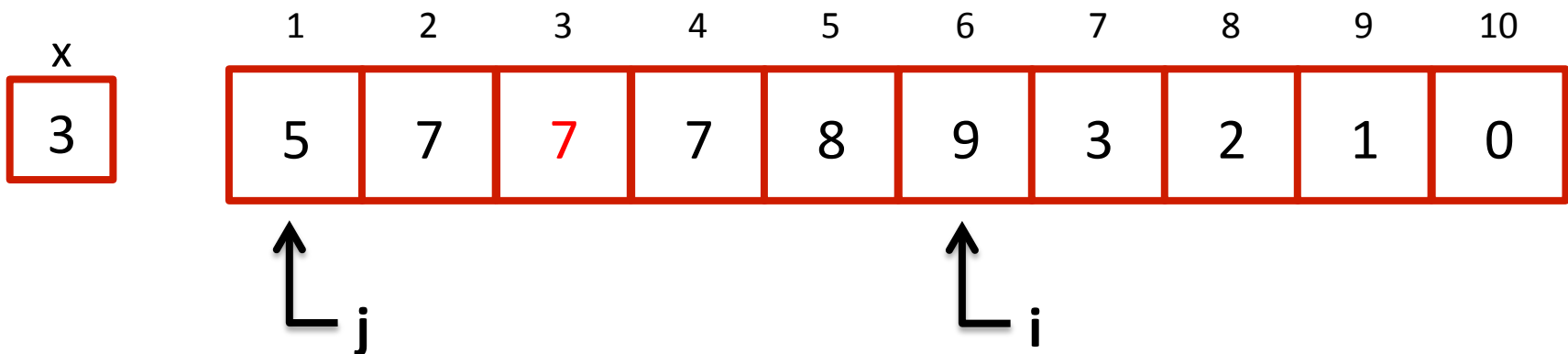
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

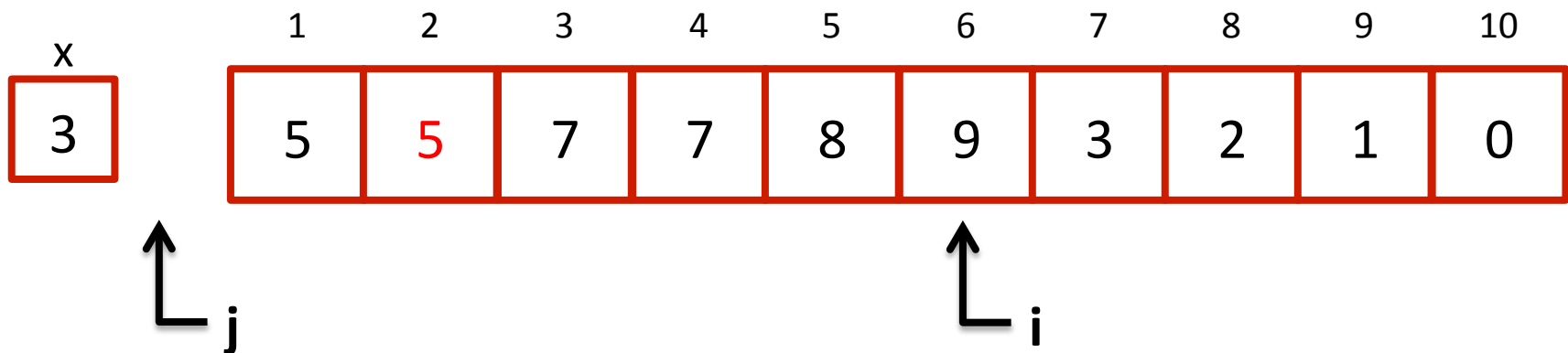
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

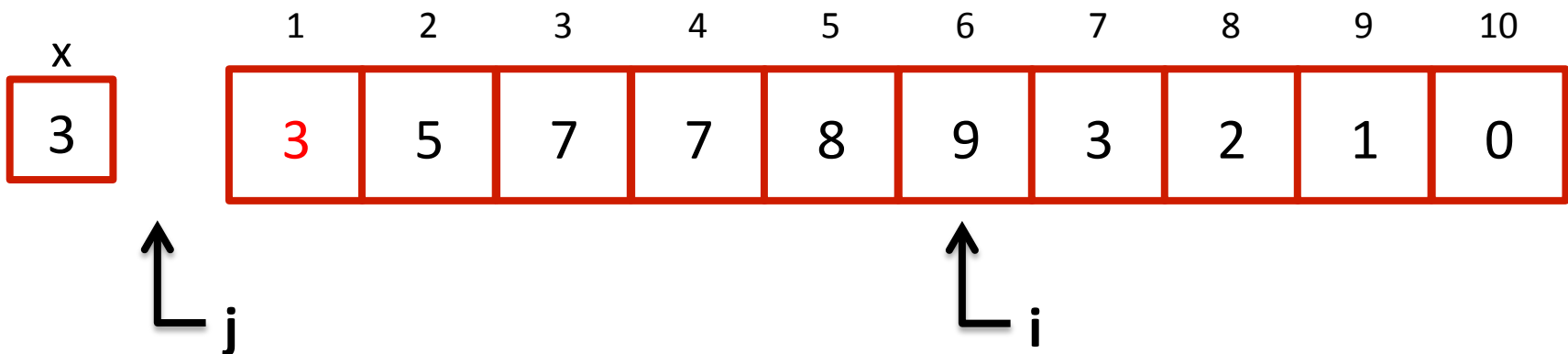
```
  end for
```

```
end procedure
```



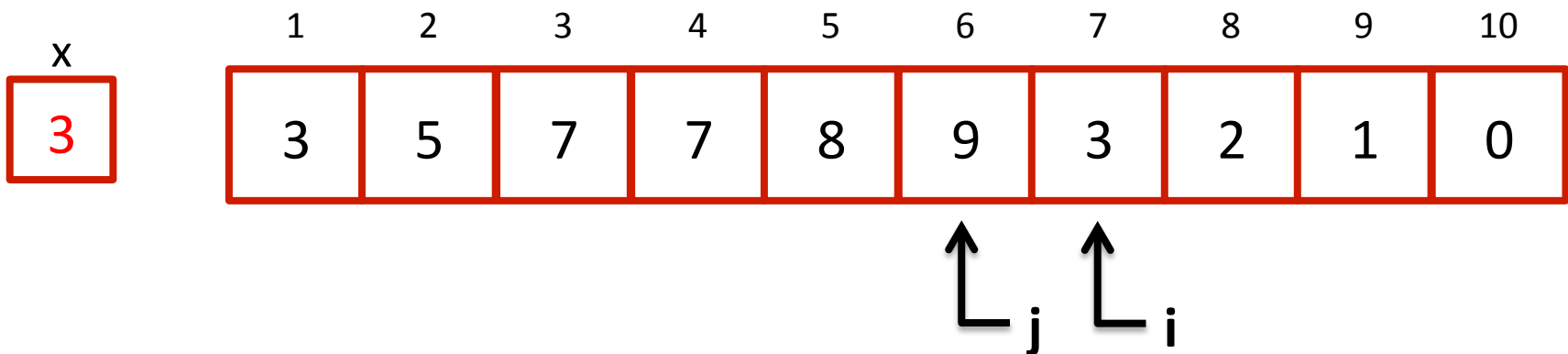
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



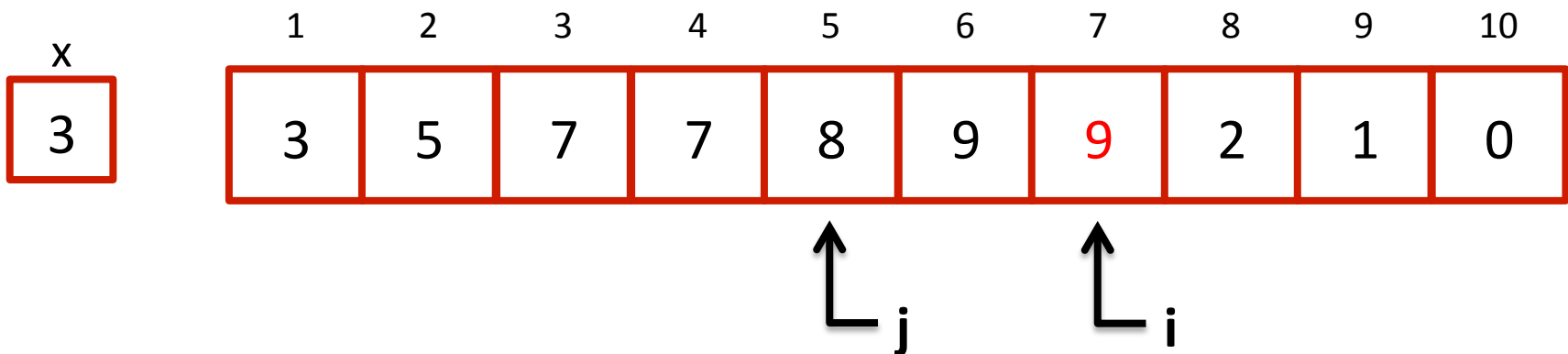
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



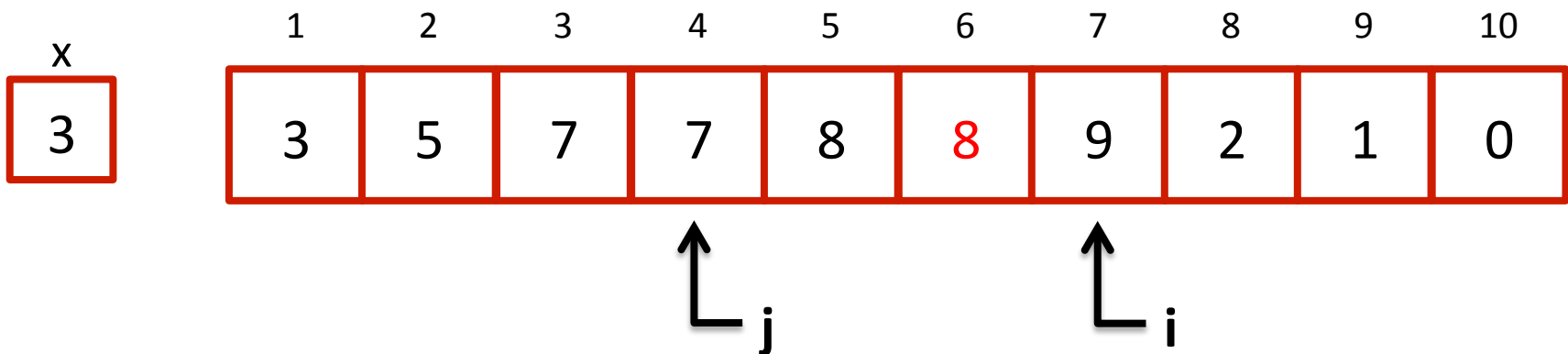
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



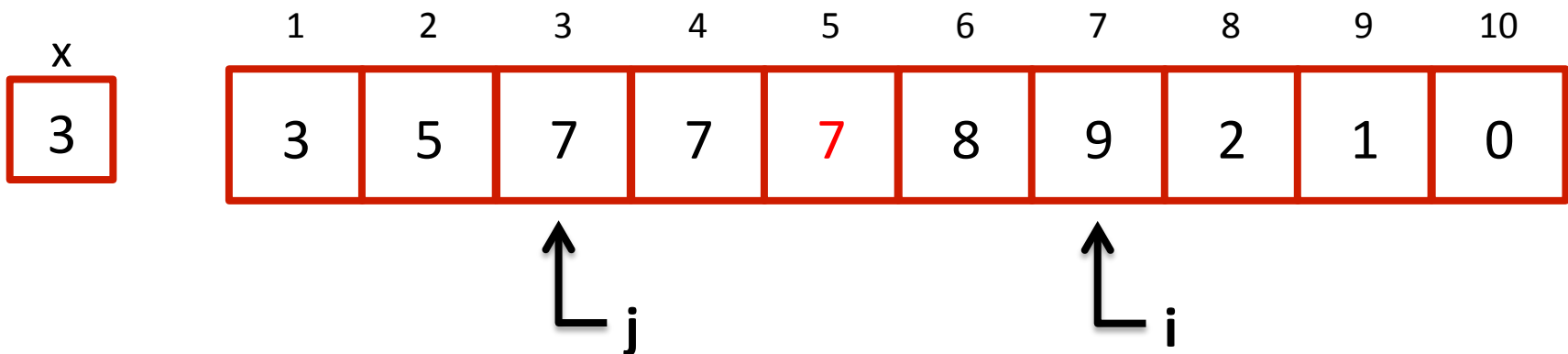
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



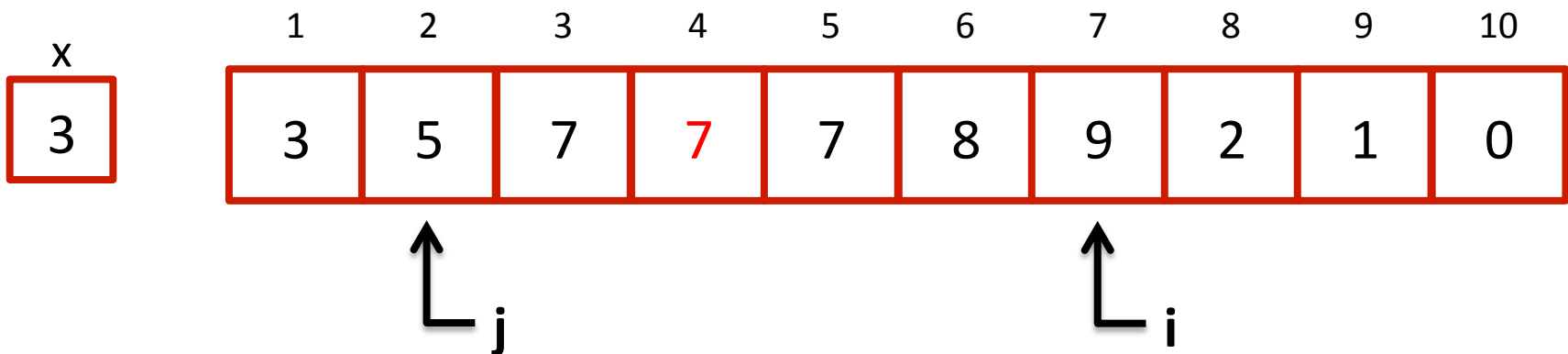
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



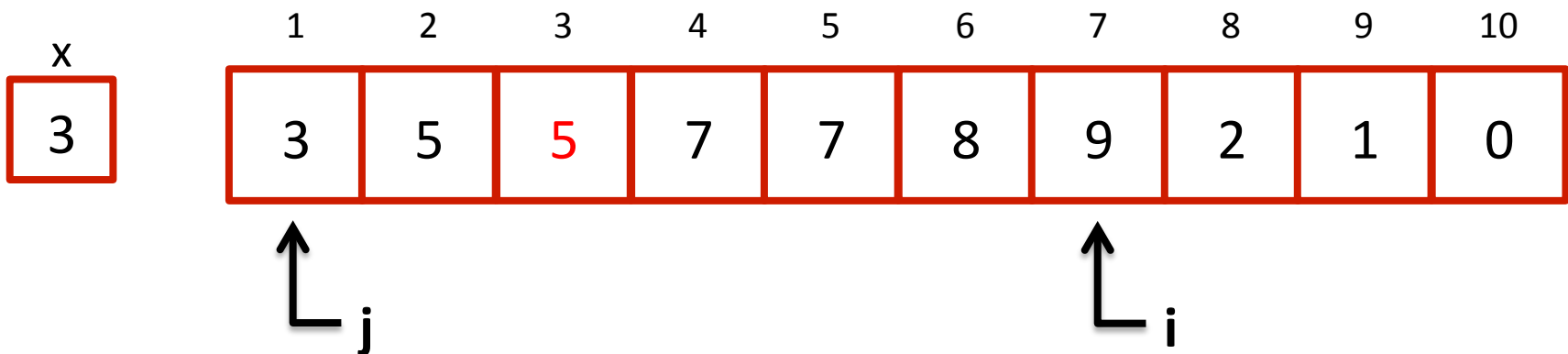
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

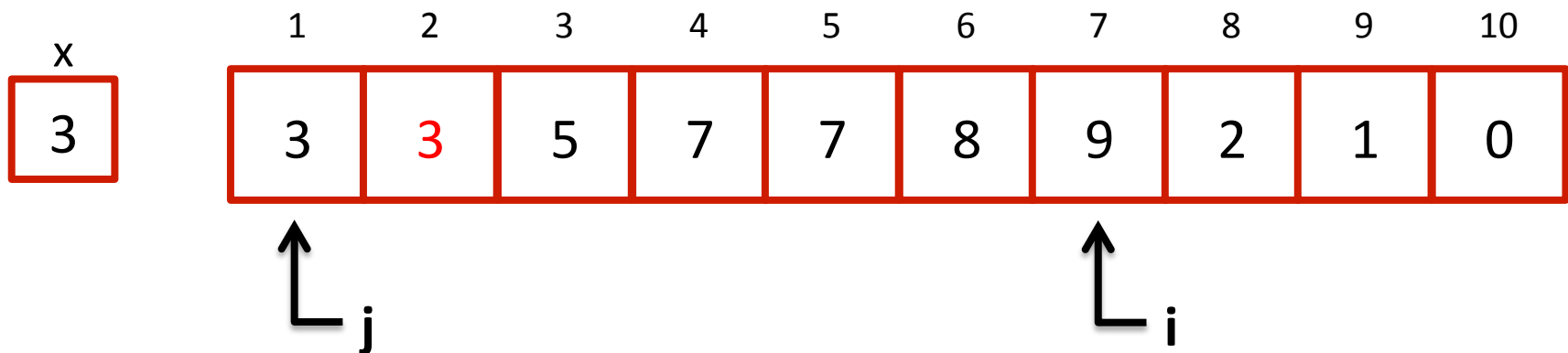
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

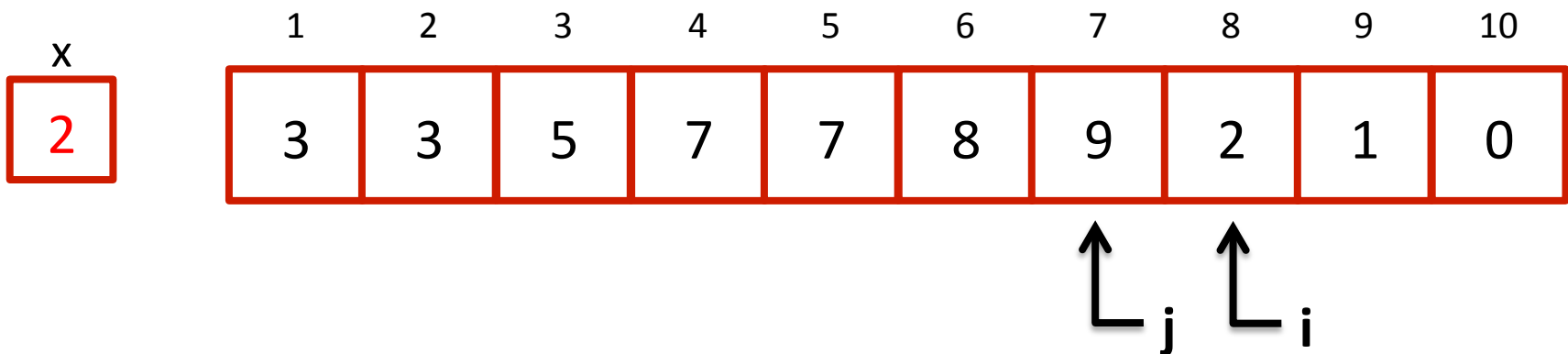
```
  end for
```

```
end procedure
```



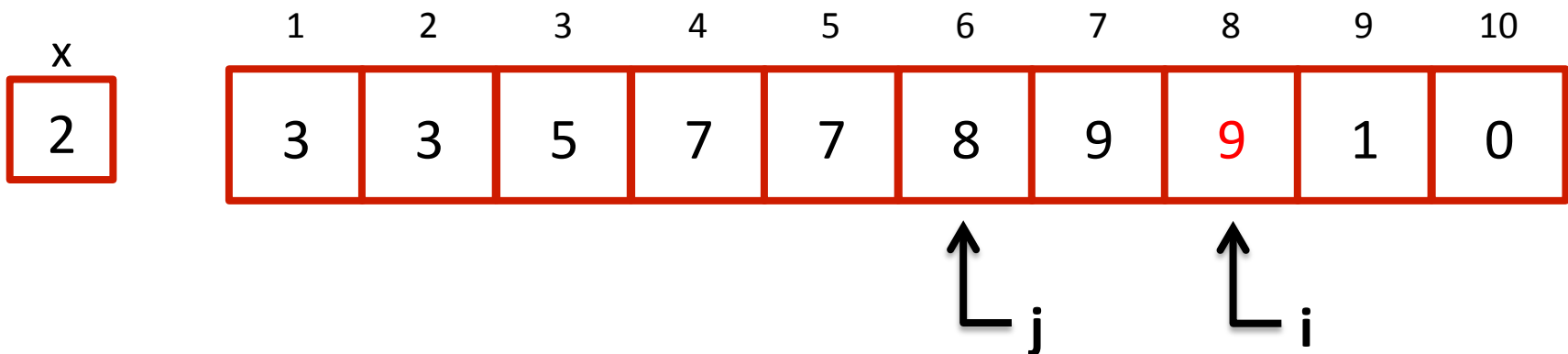
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



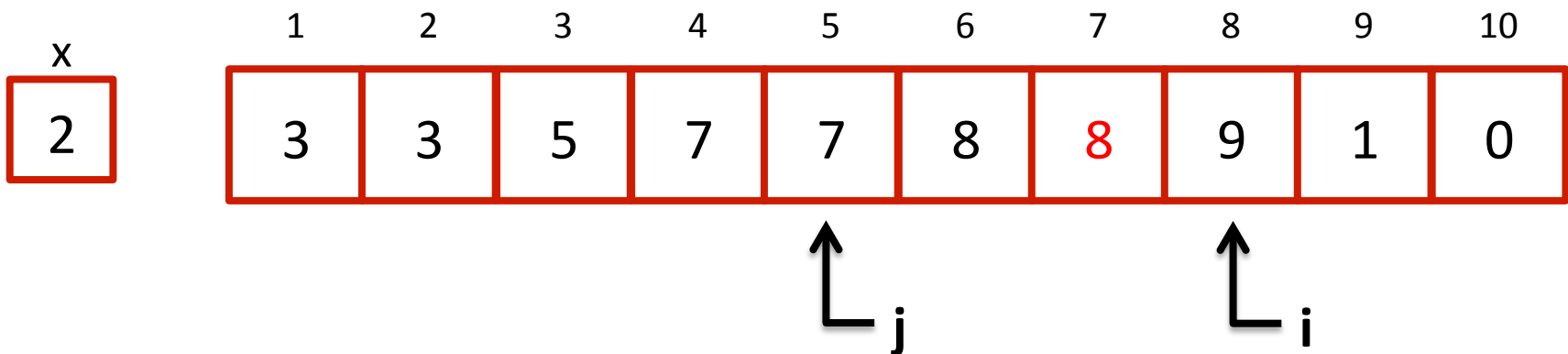
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



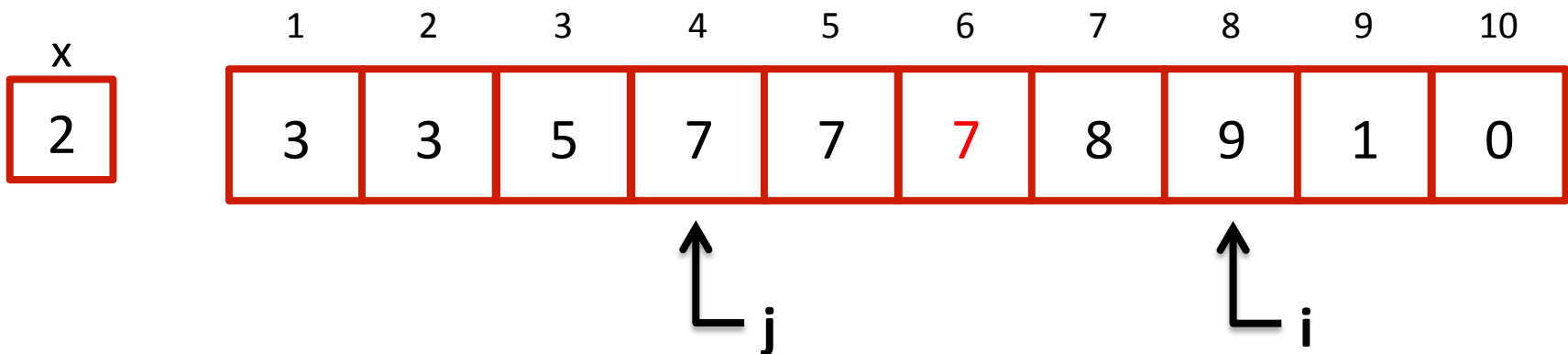
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



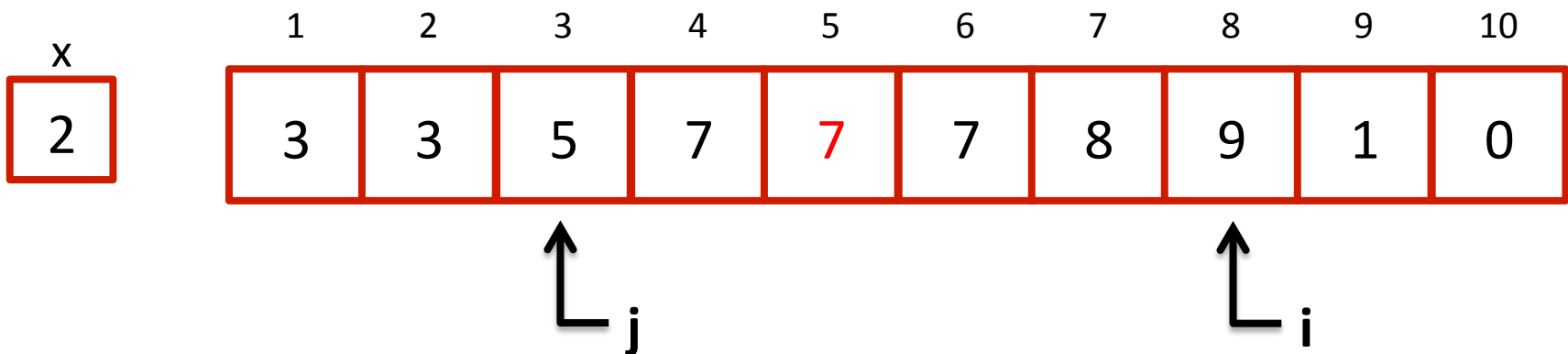
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



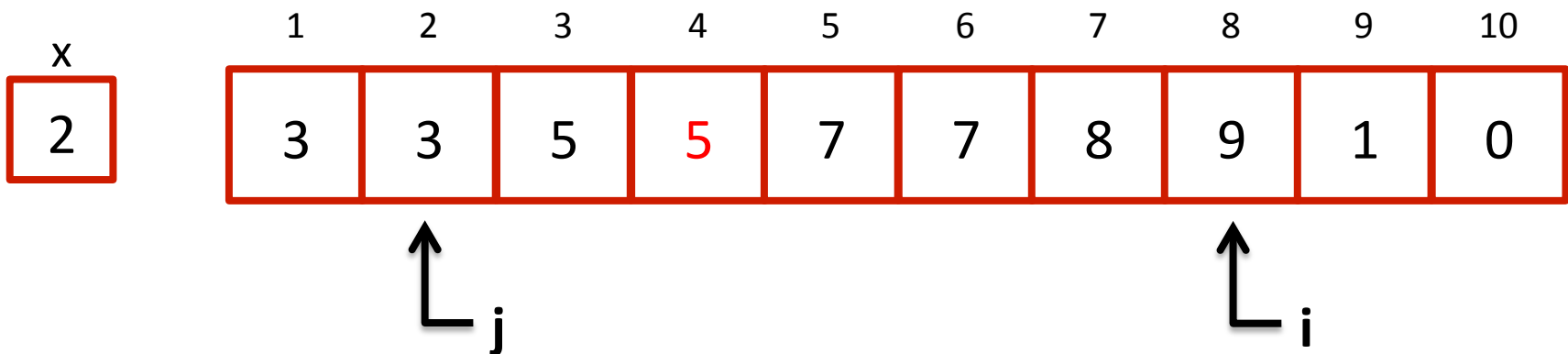
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



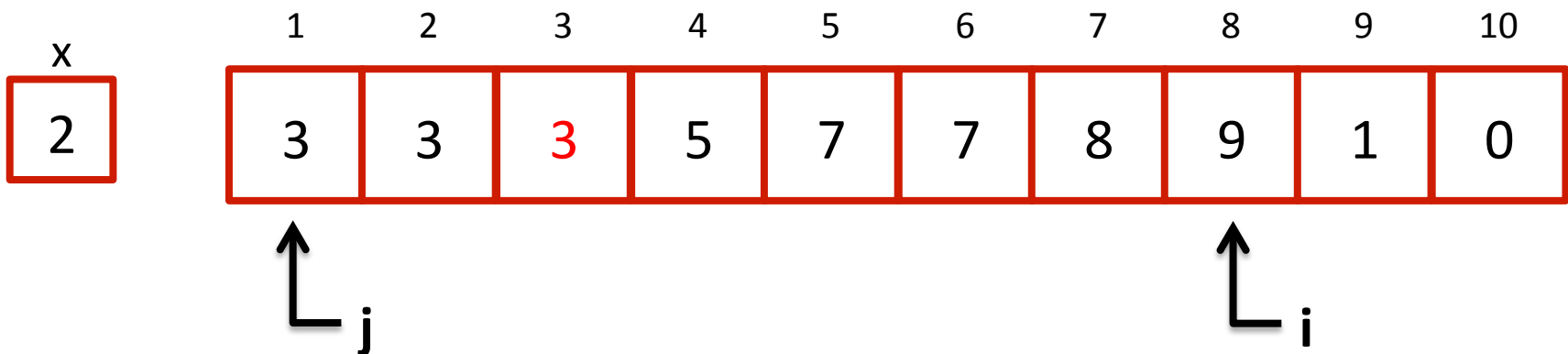
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



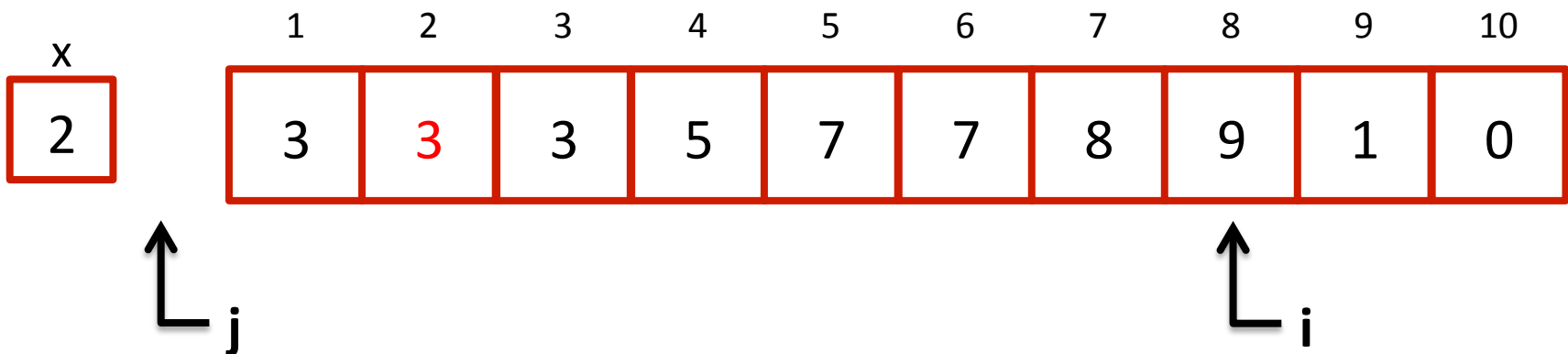
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



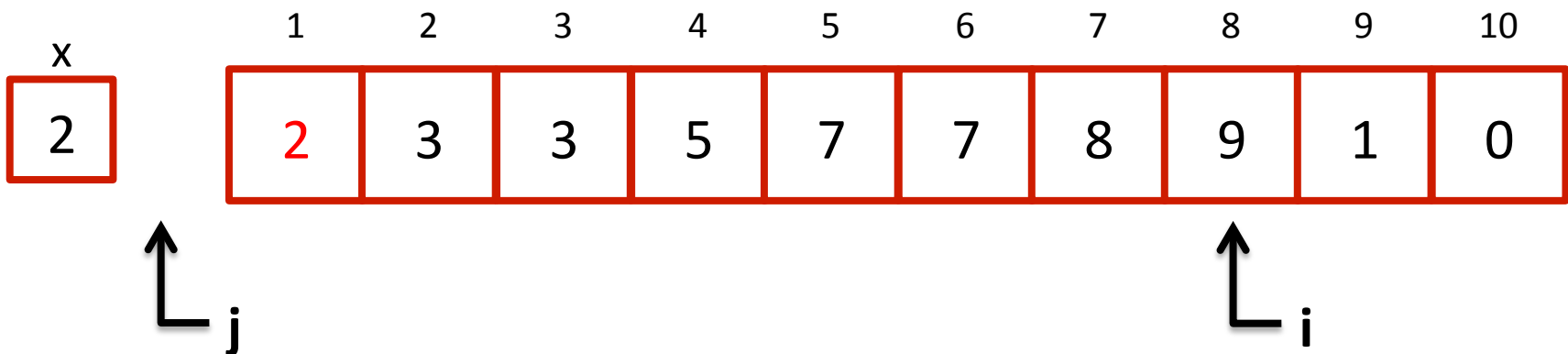
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



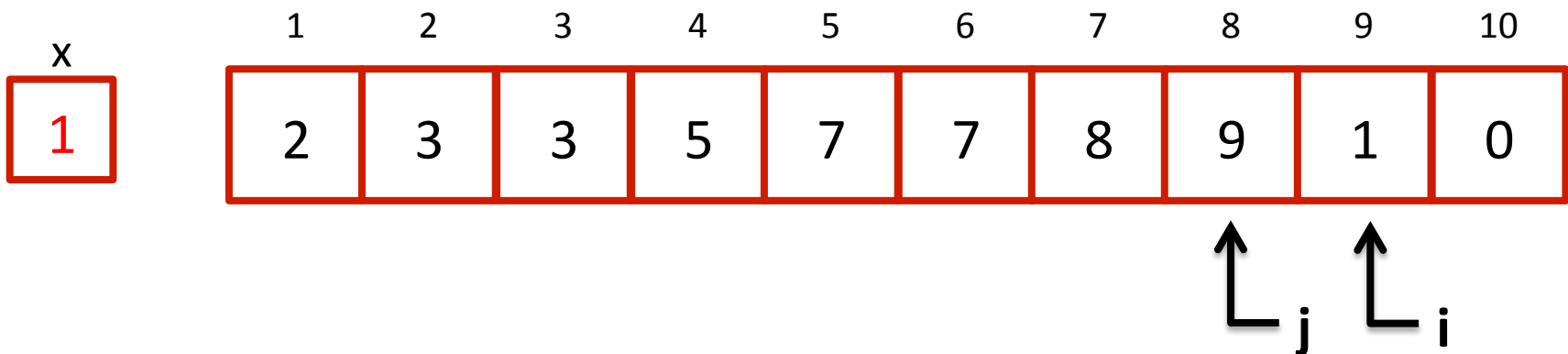
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



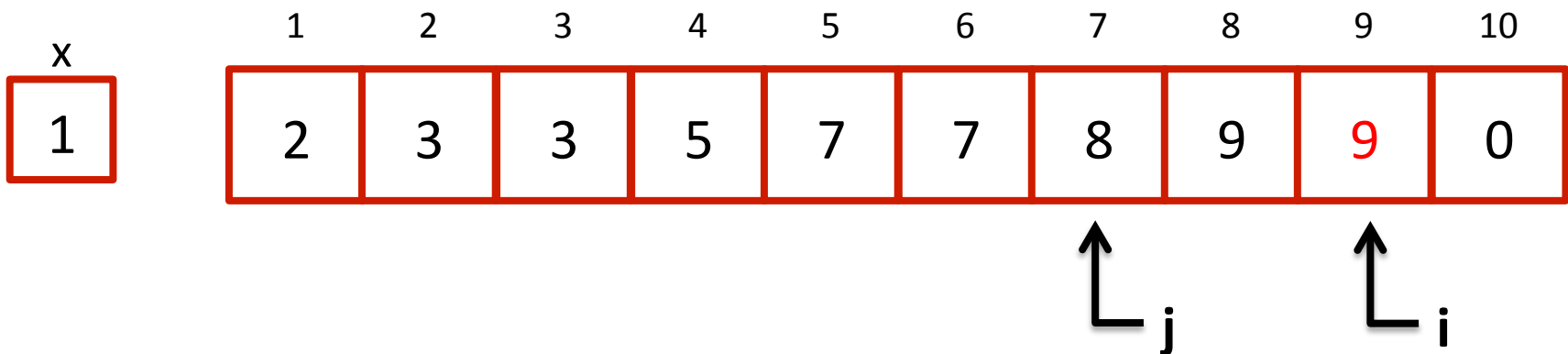
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



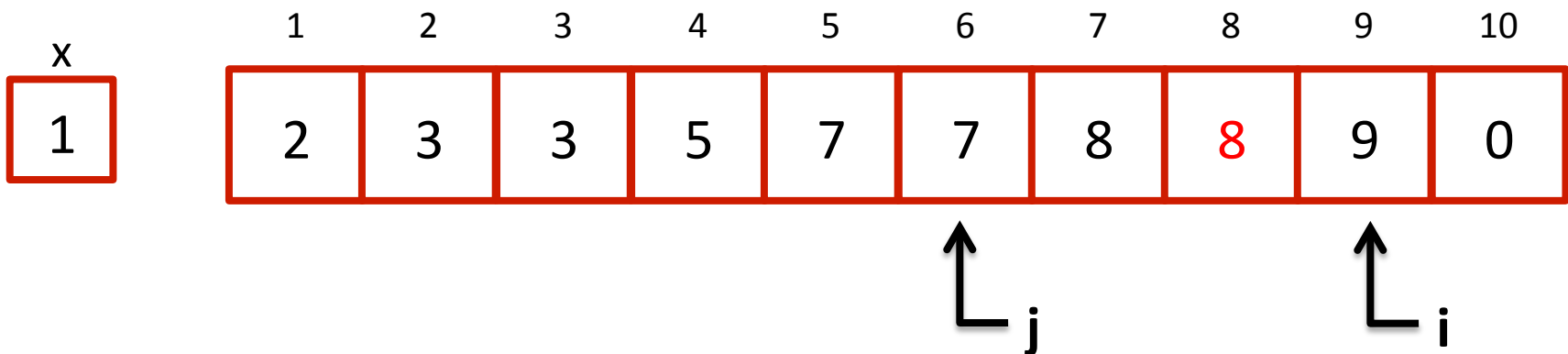
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



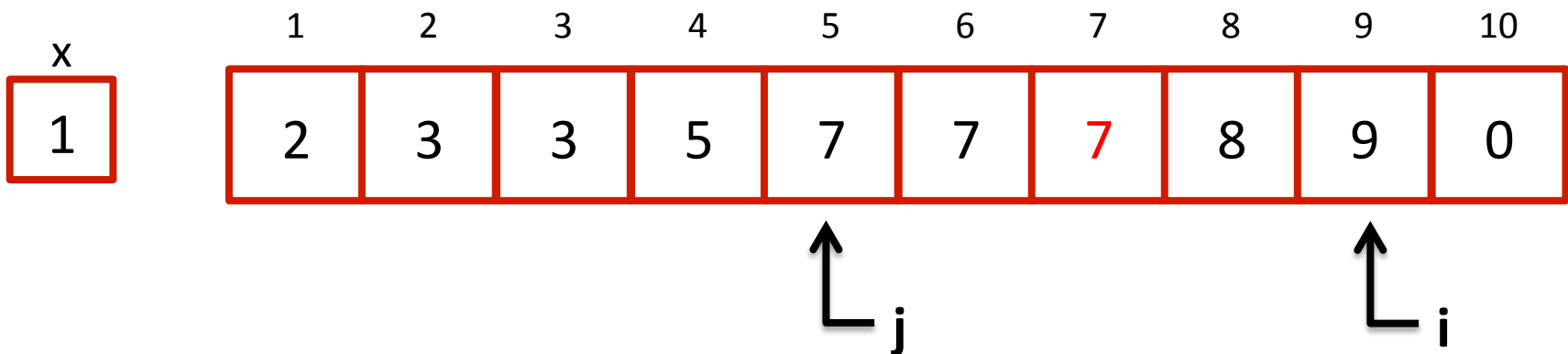
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



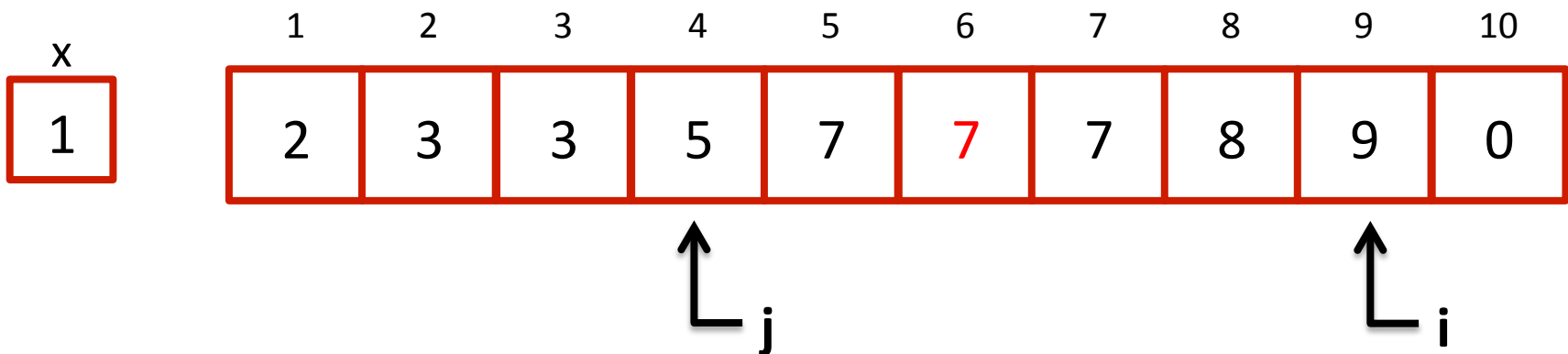
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



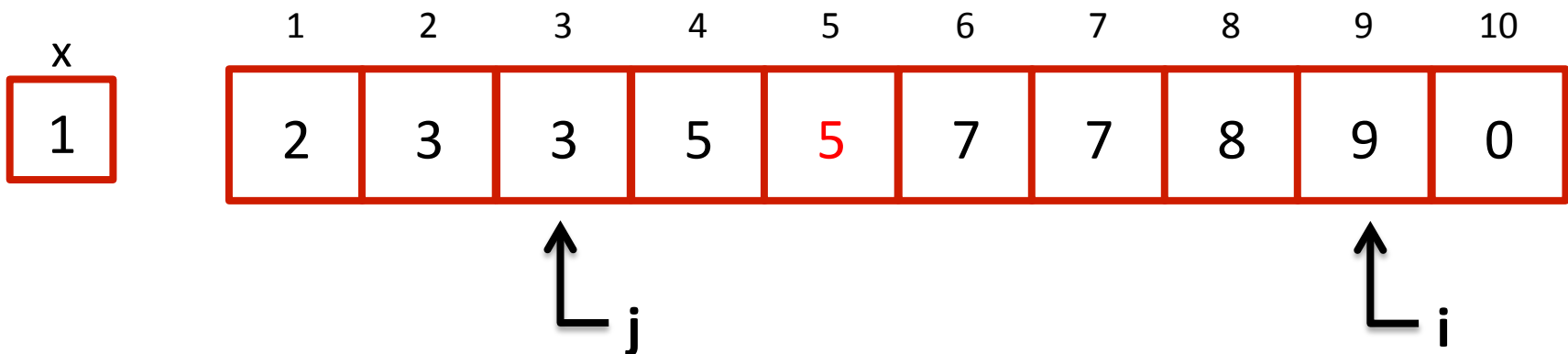
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



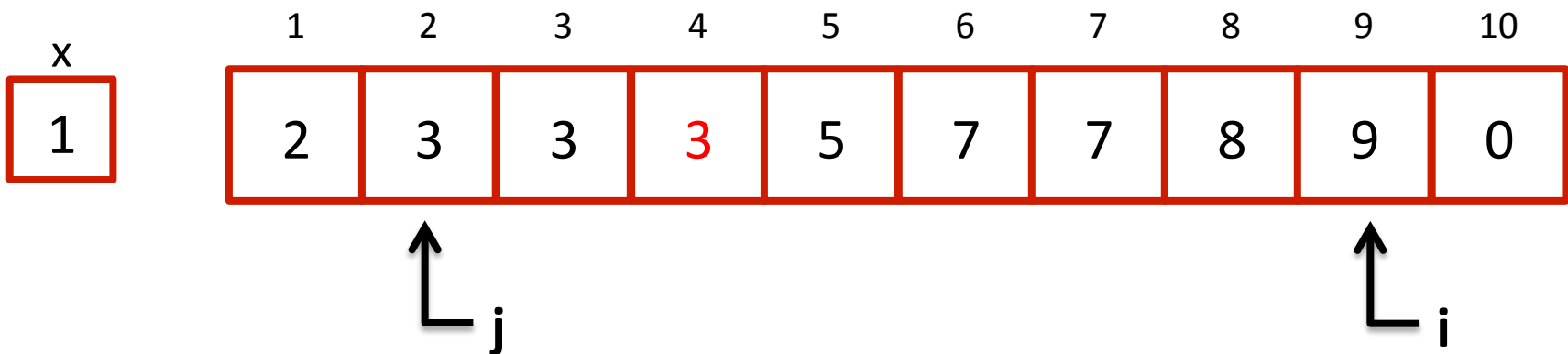
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



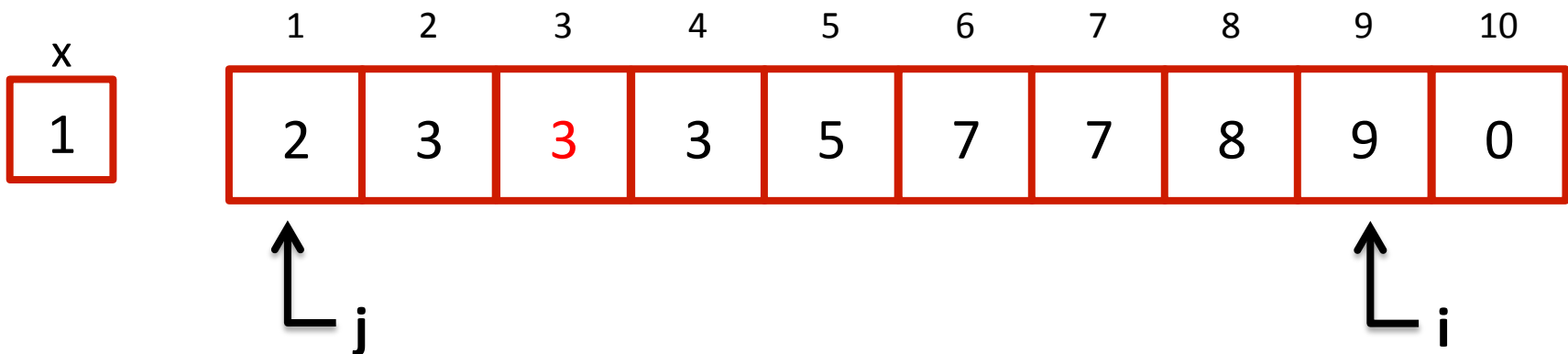
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



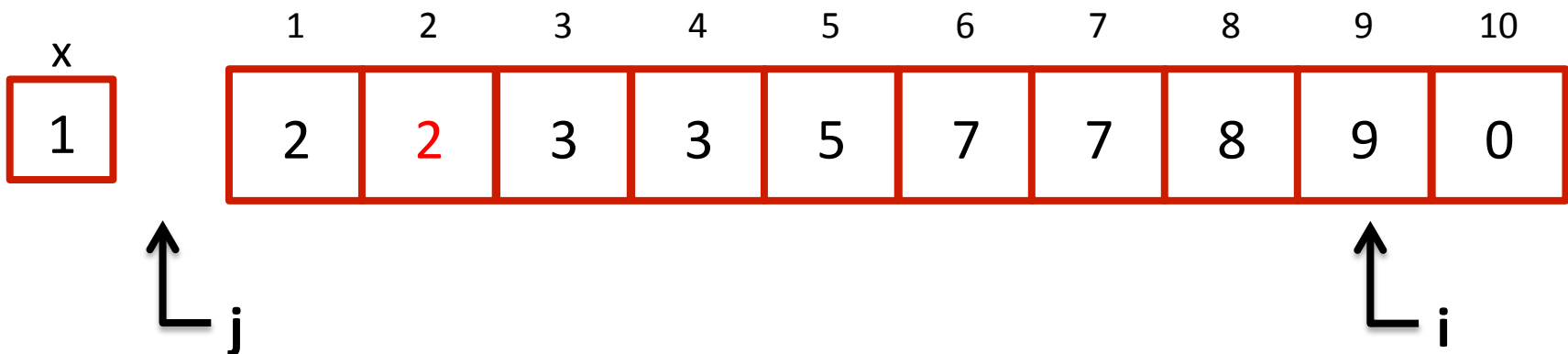
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

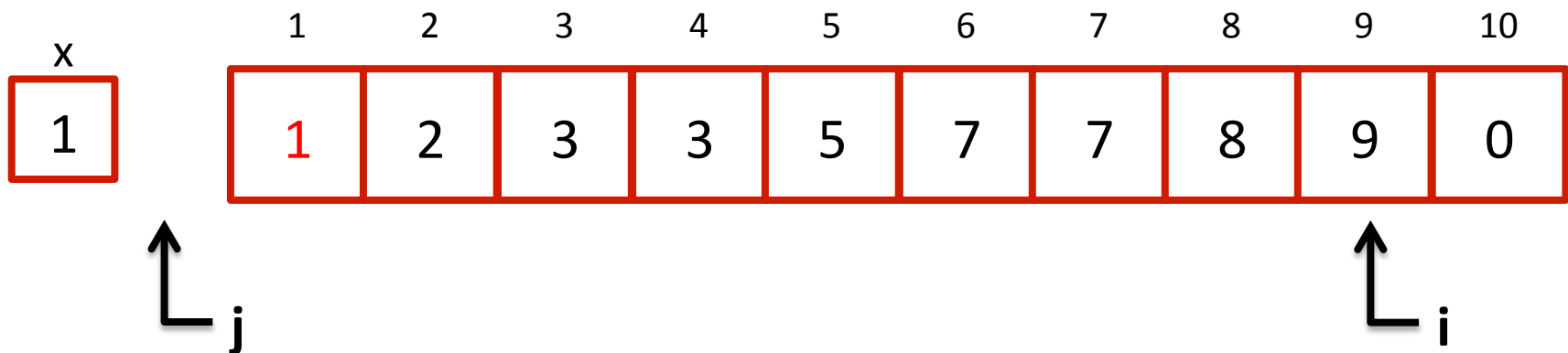
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

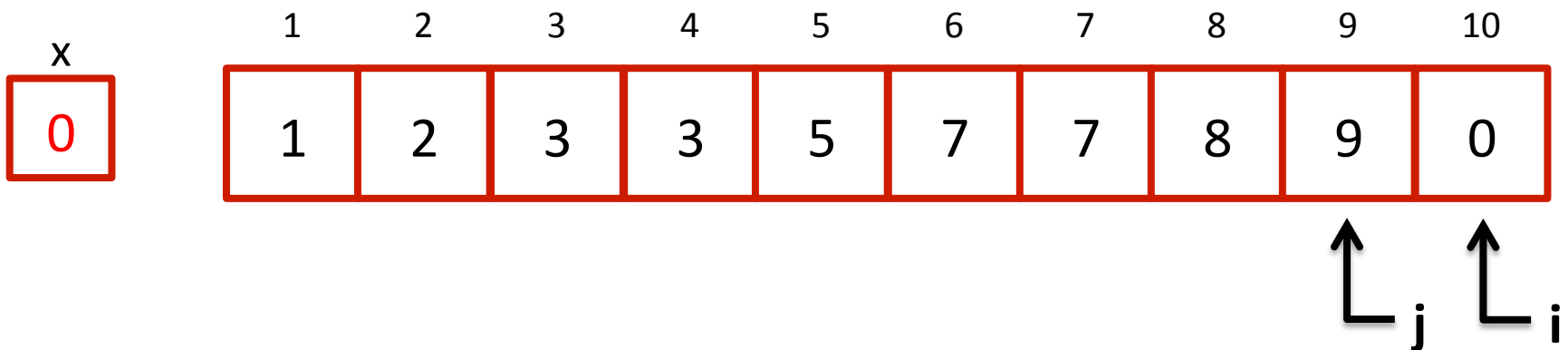
```
  end for
```

```
end procedure
```



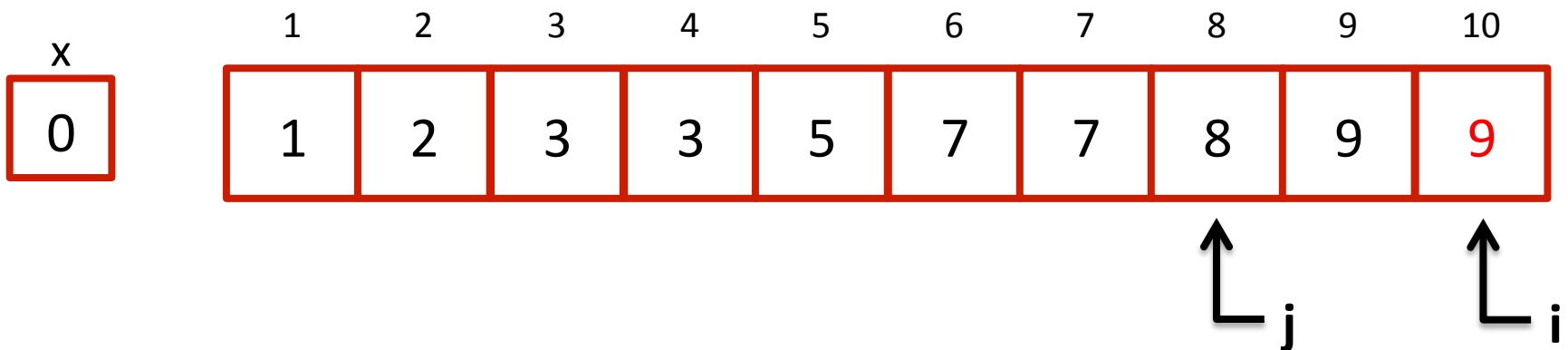
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

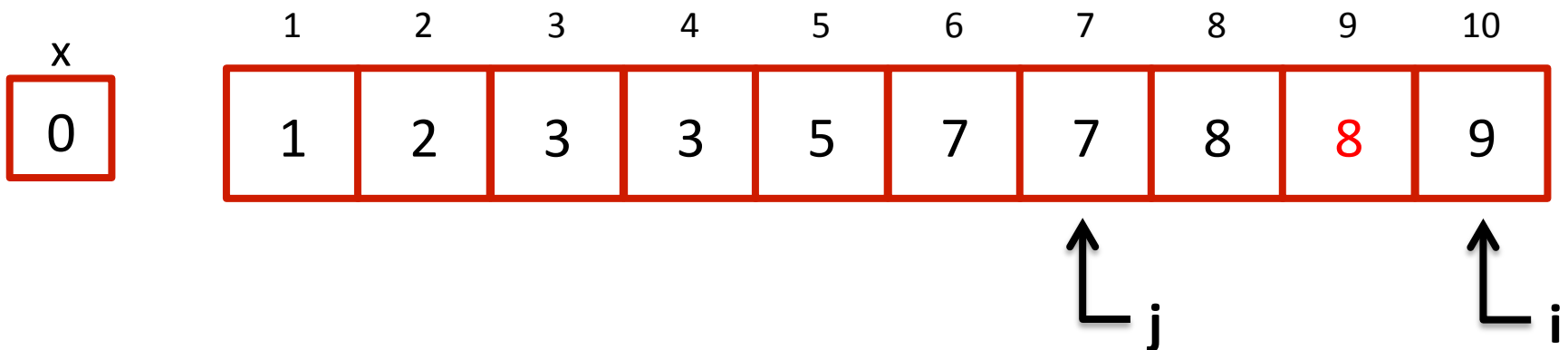
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

```
  end for
```

```
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

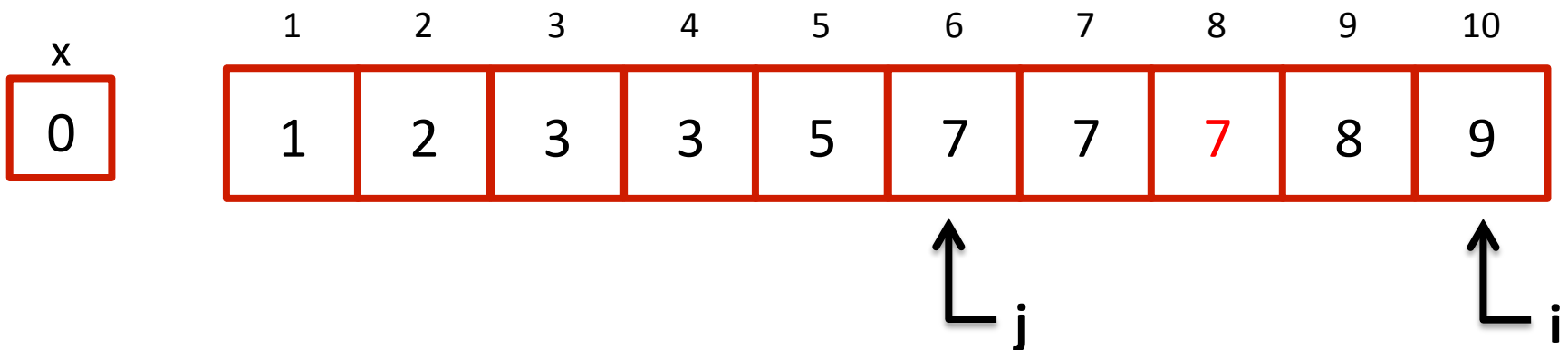
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

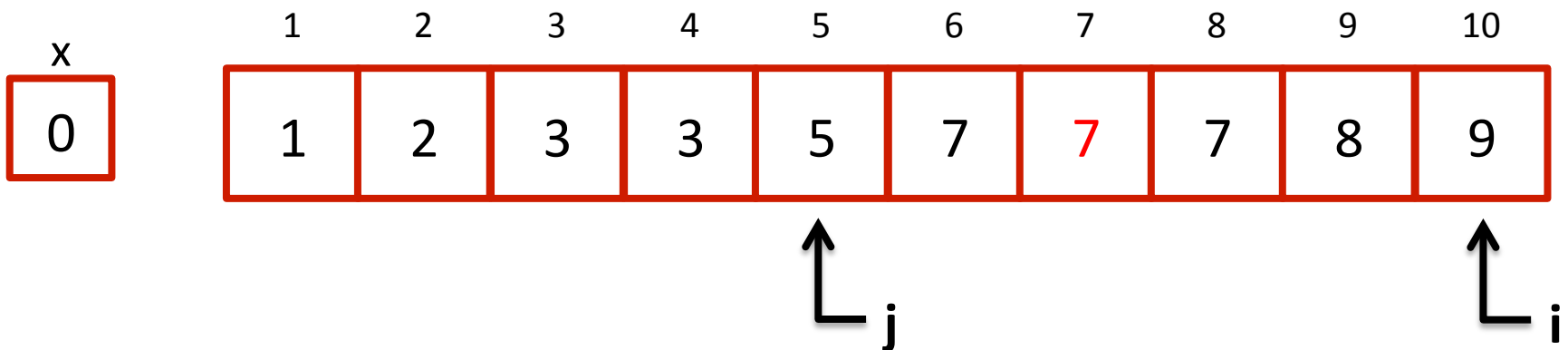
```
  end for
```

```
end procedure
```



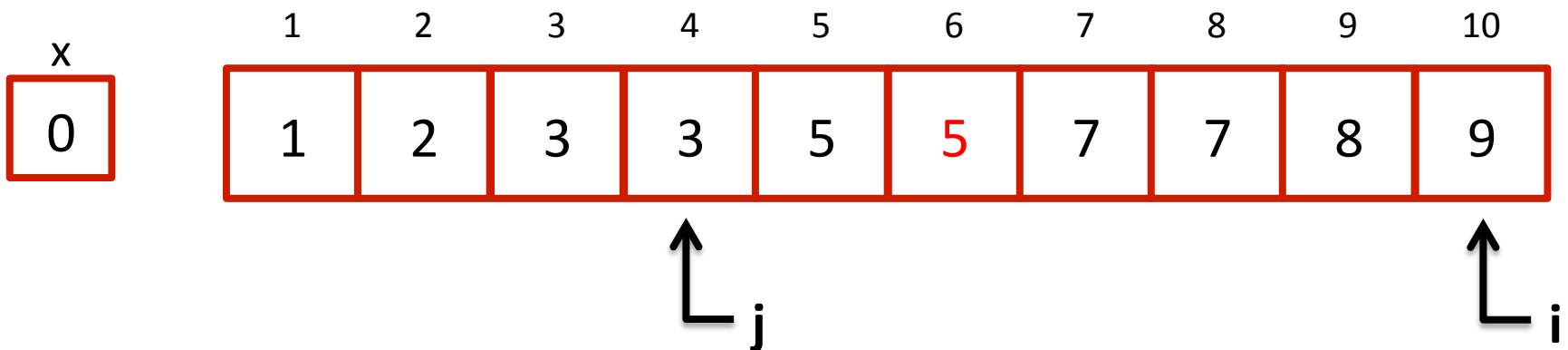
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



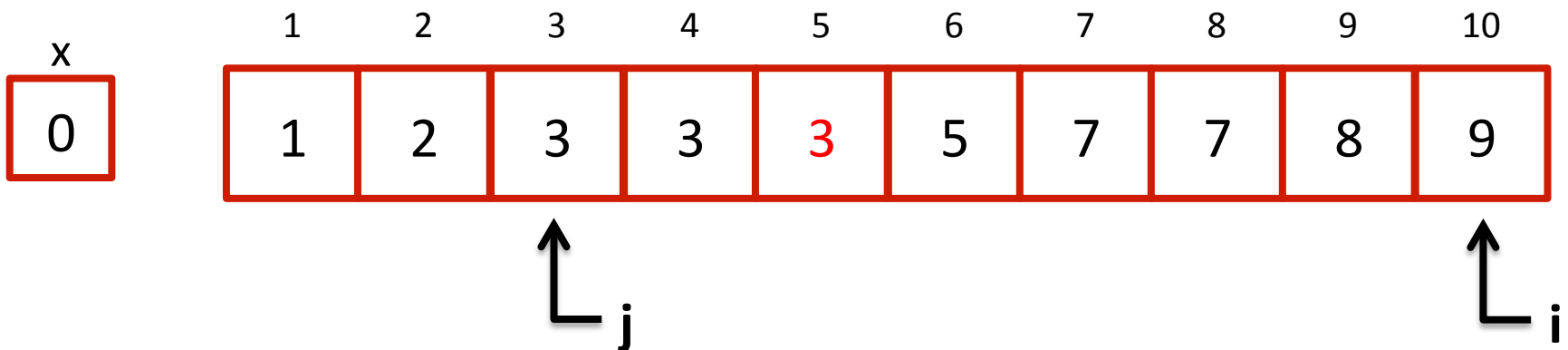
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



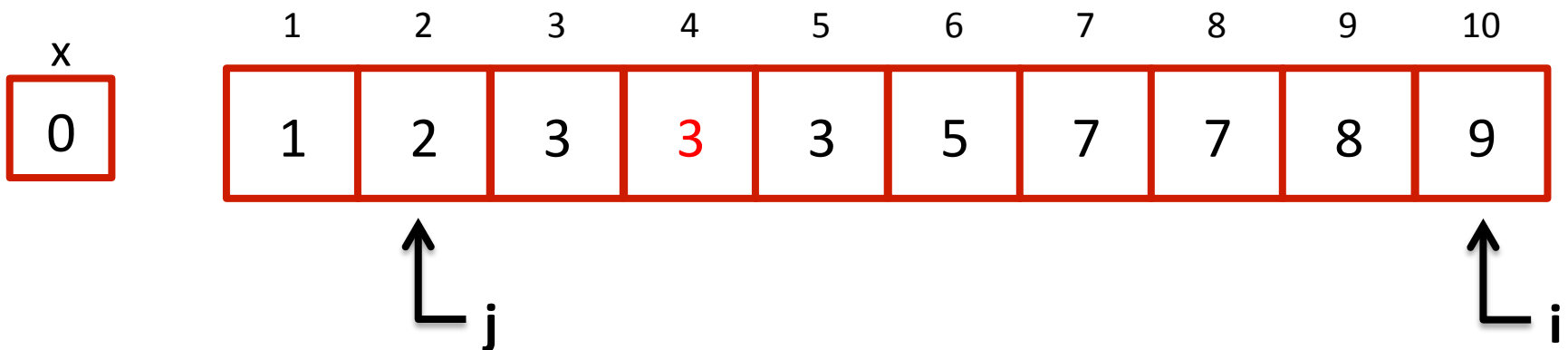
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



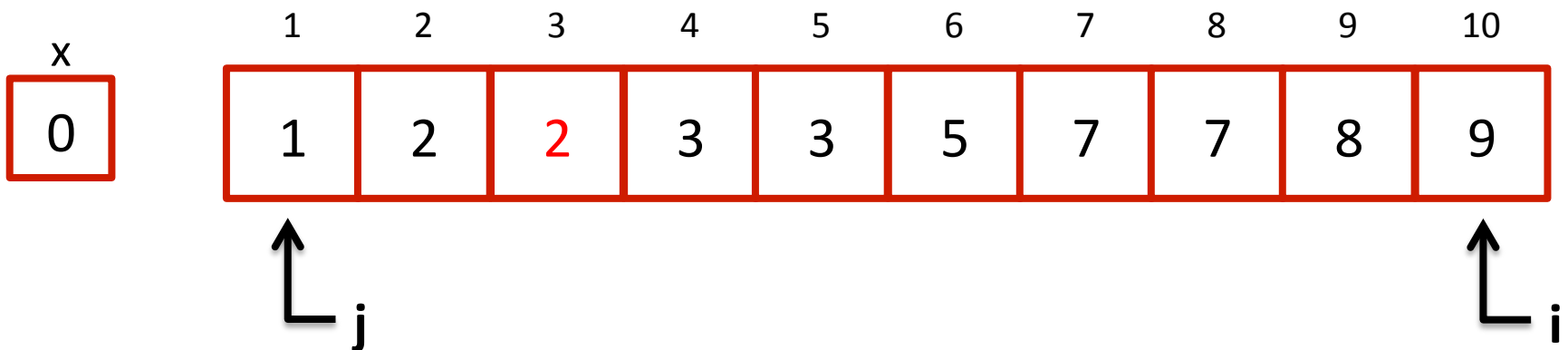
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



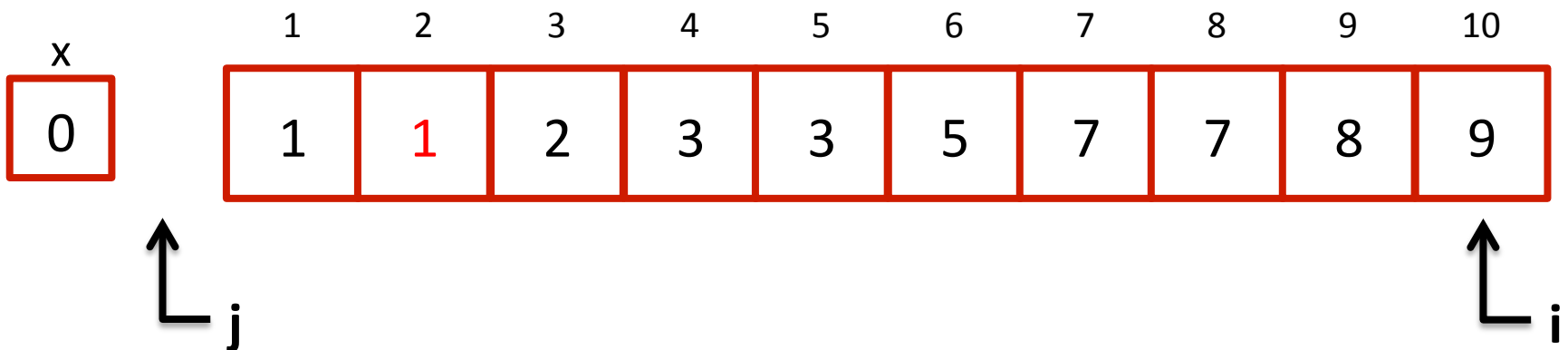
Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

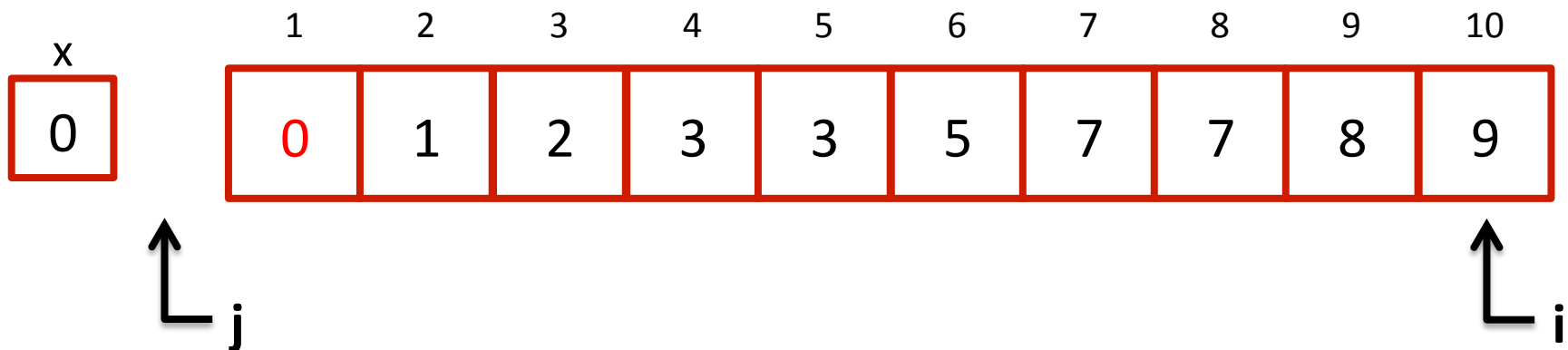
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

```
  end for
```

```
end procedure
```



Insertion sort: Worst case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

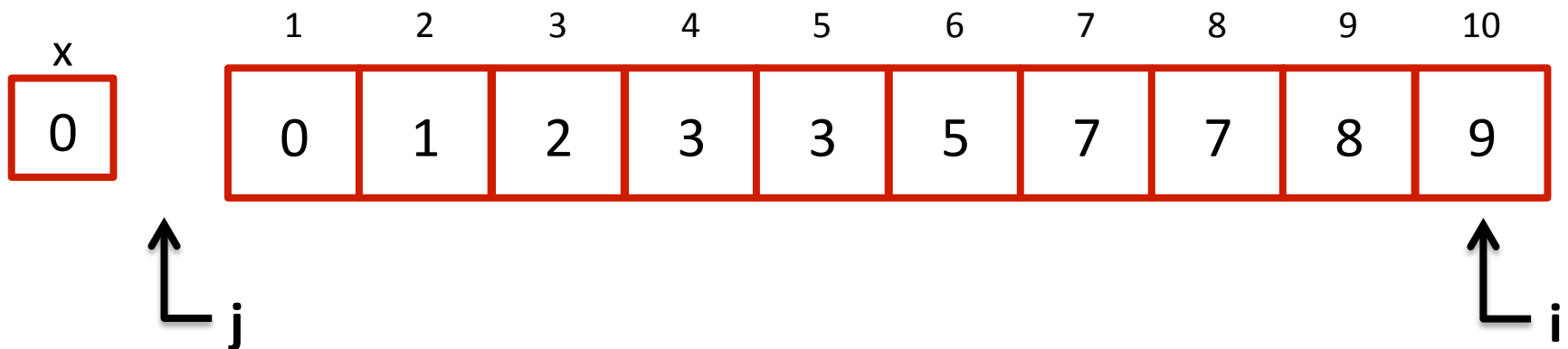
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

```
  end for
```

```
end procedure
```



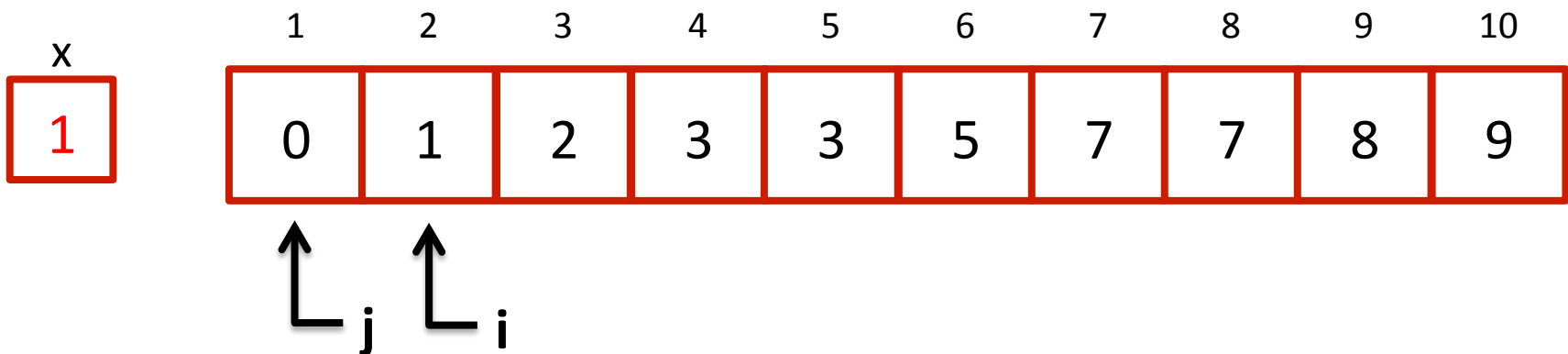
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```

1	2	3	4	5	6	7	8	9	10
0	1	2	3	3	5	7	7	8	9

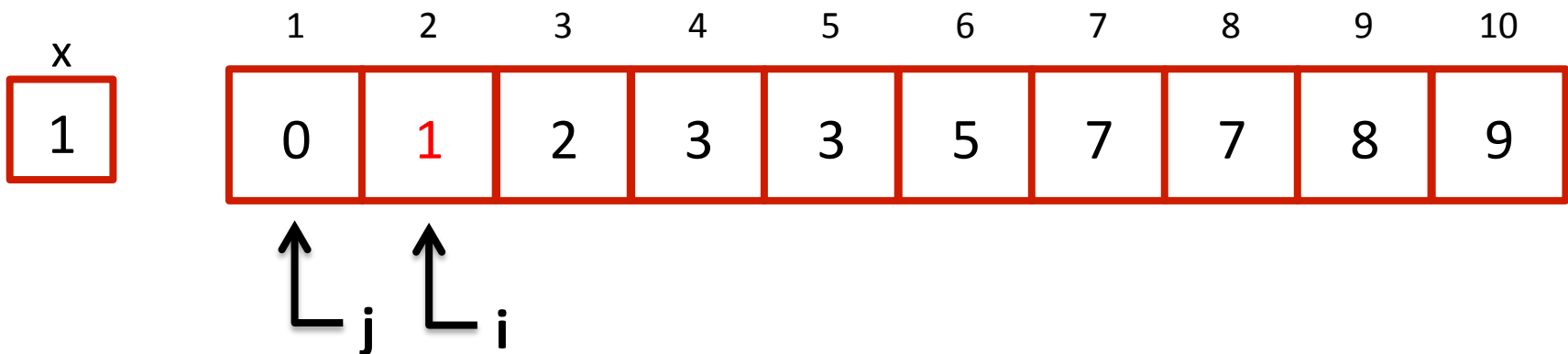
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



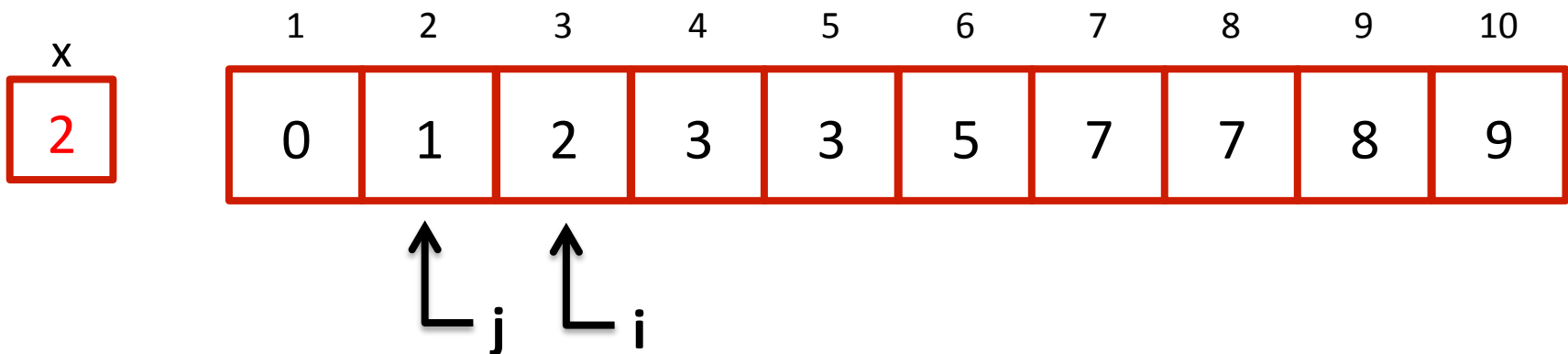
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



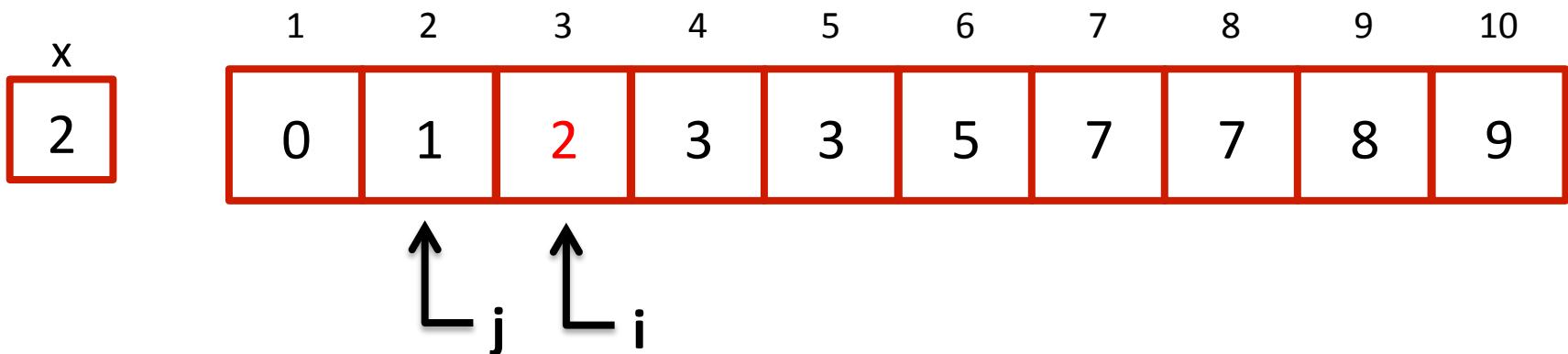
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



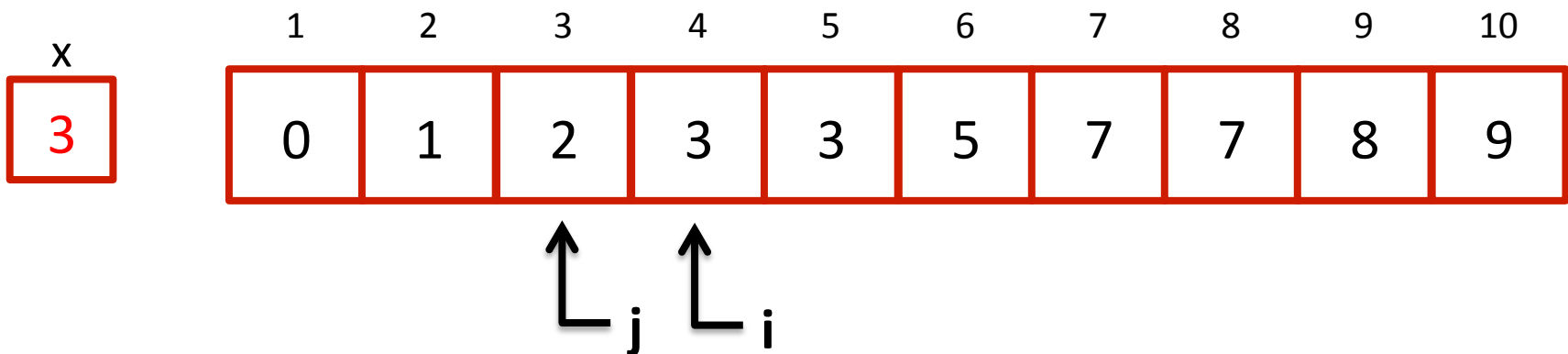
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



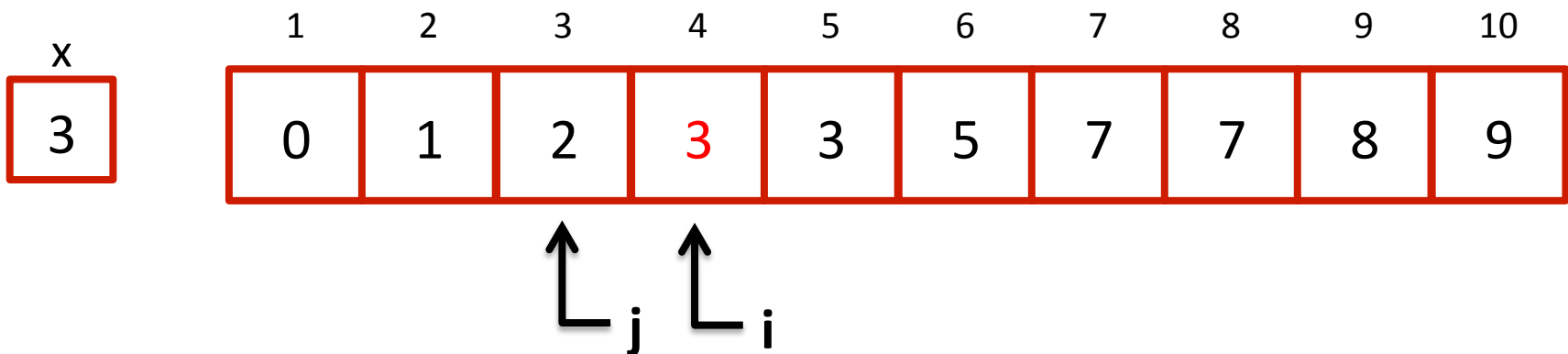
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



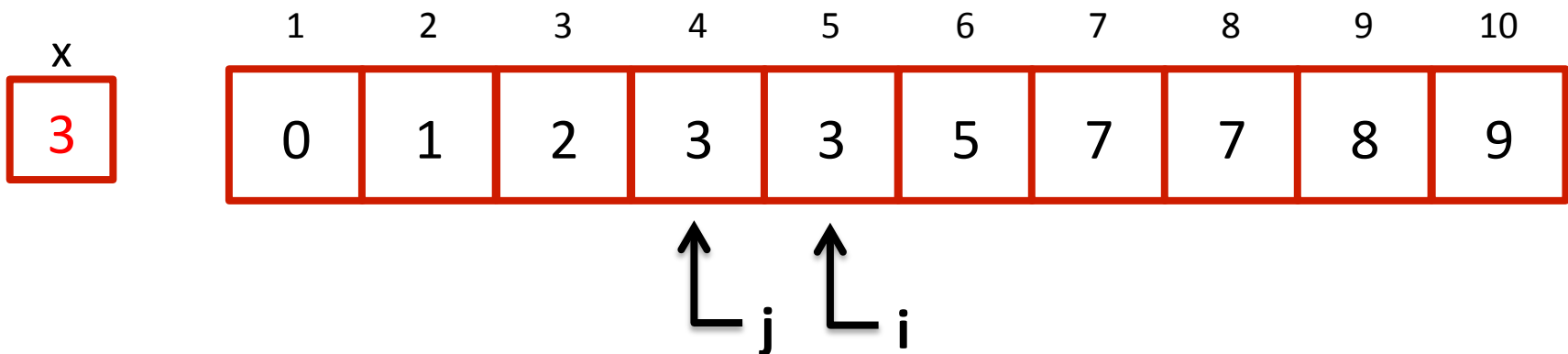
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



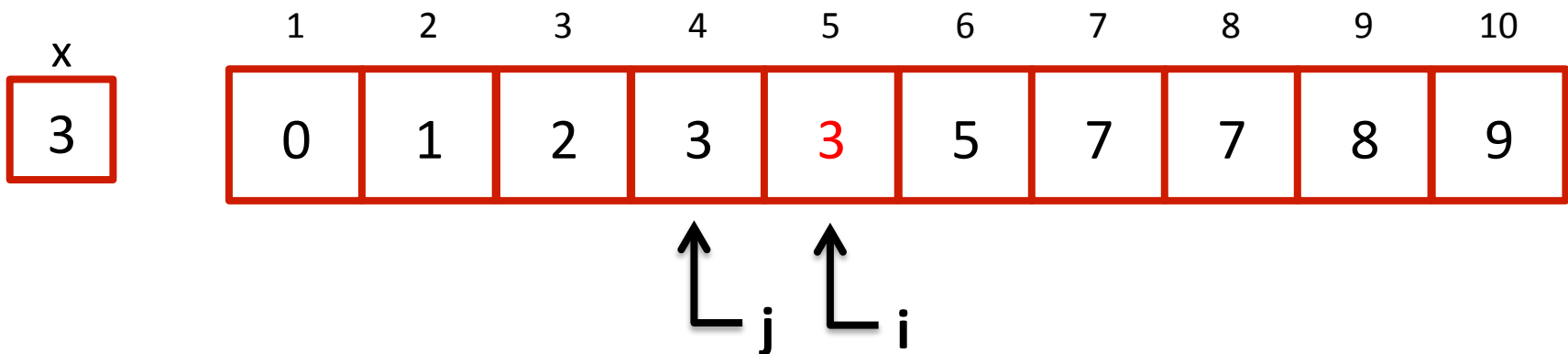
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



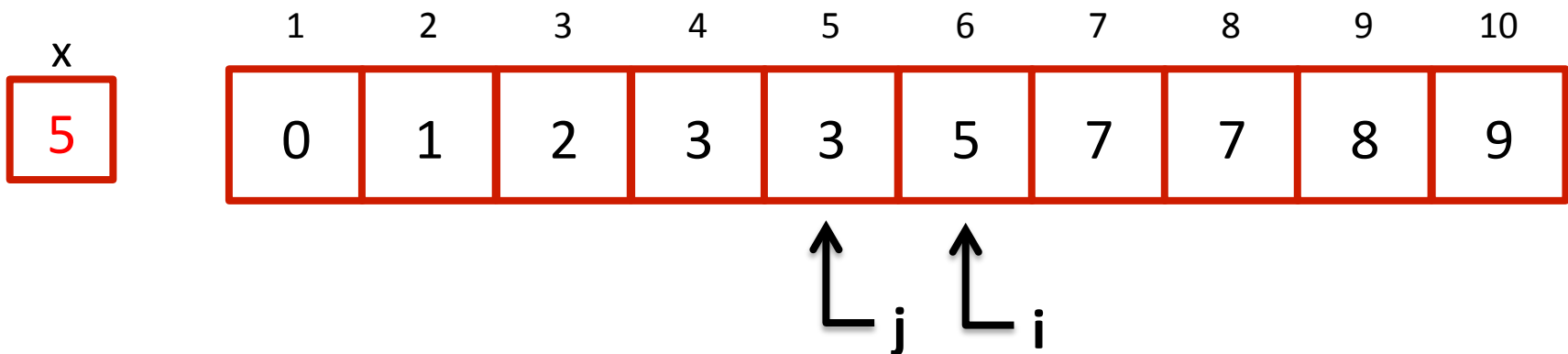
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



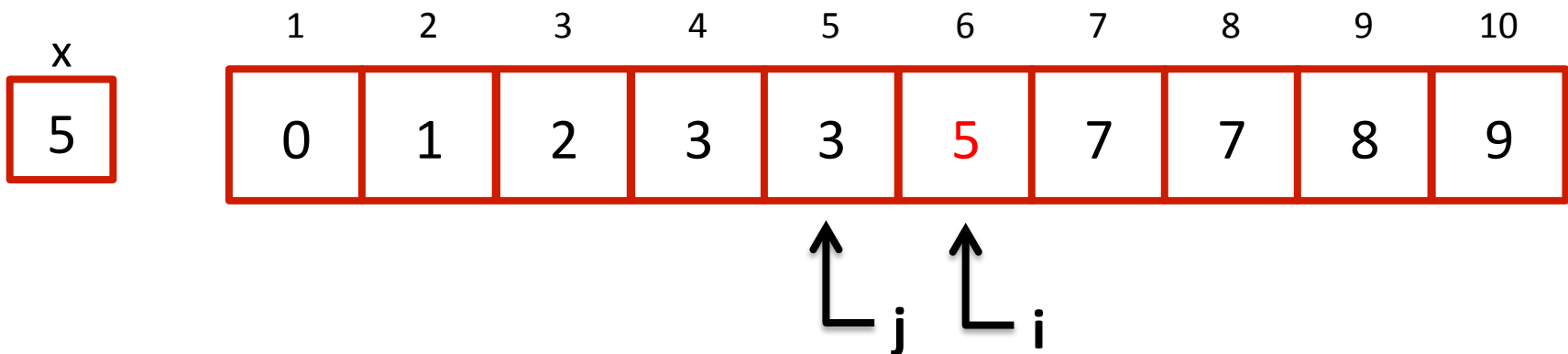
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



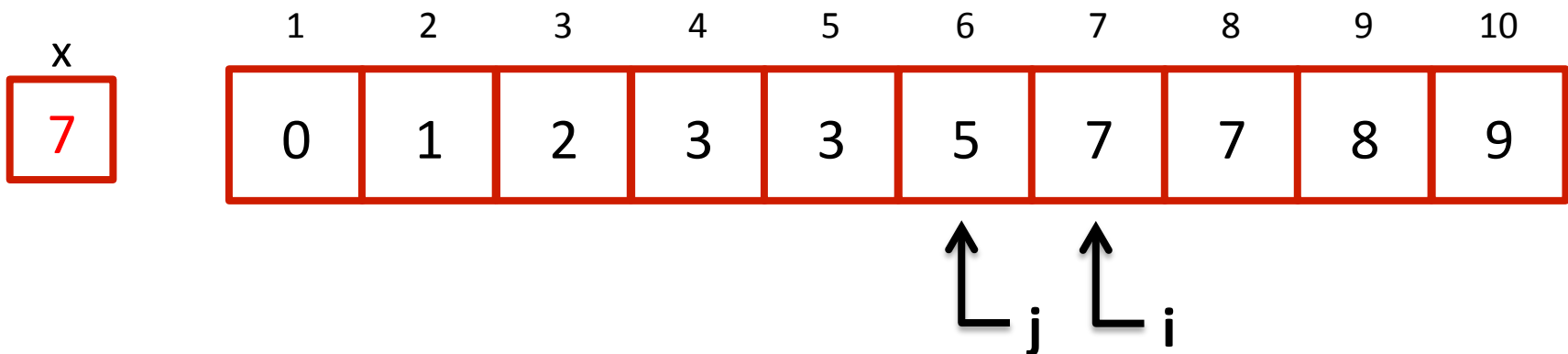
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



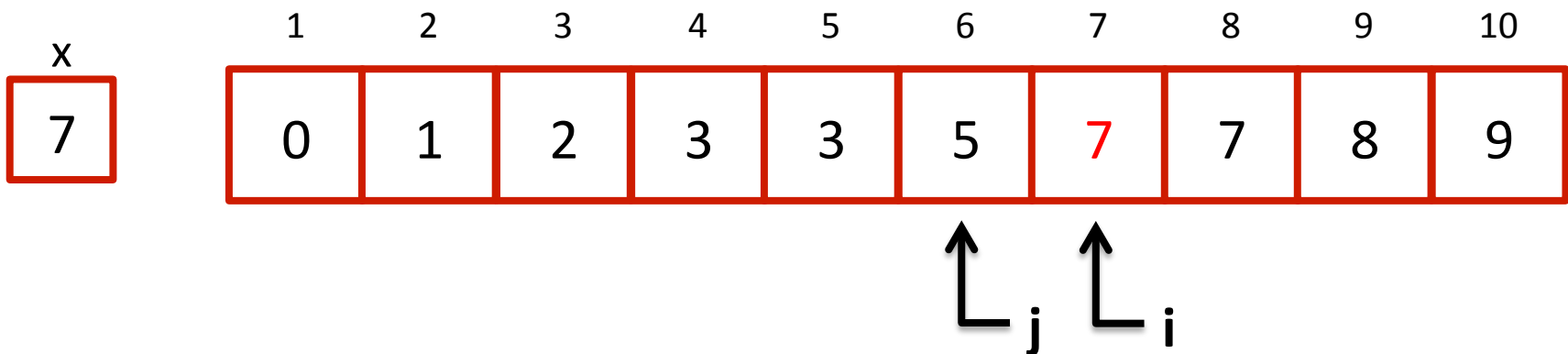
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



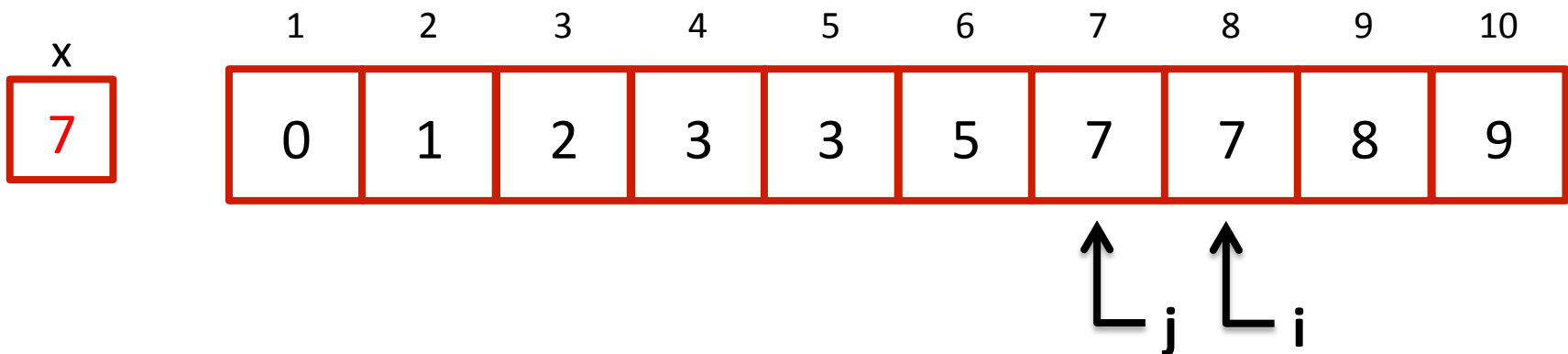
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



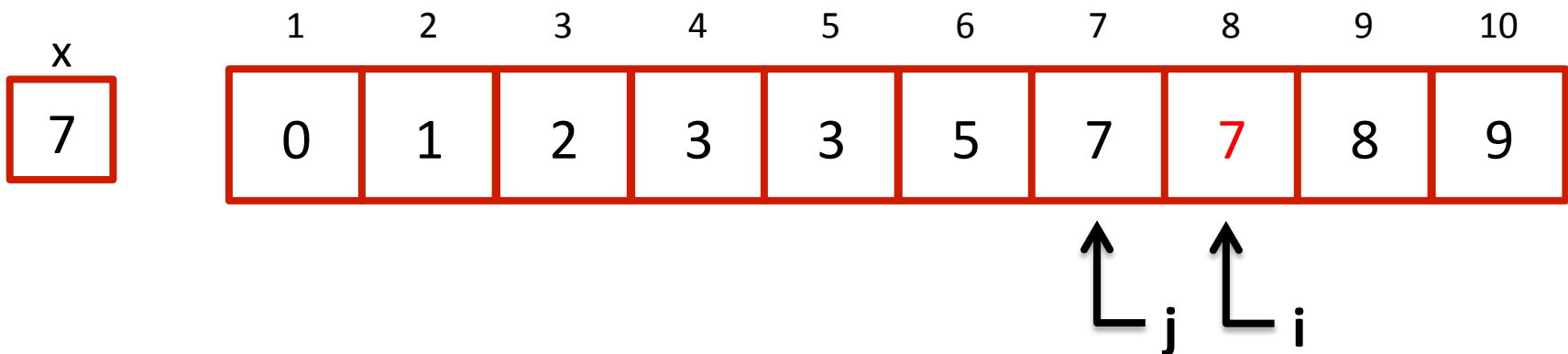
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



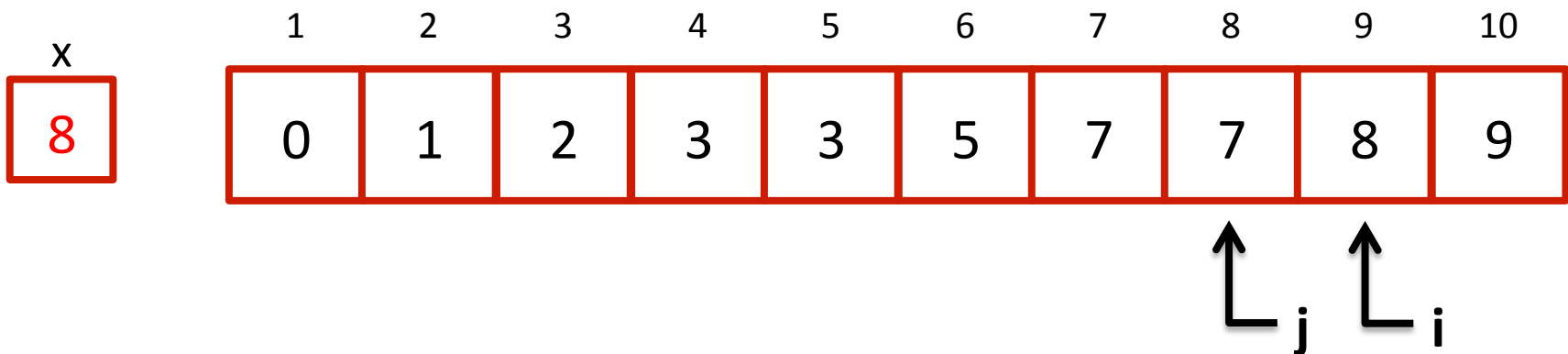
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



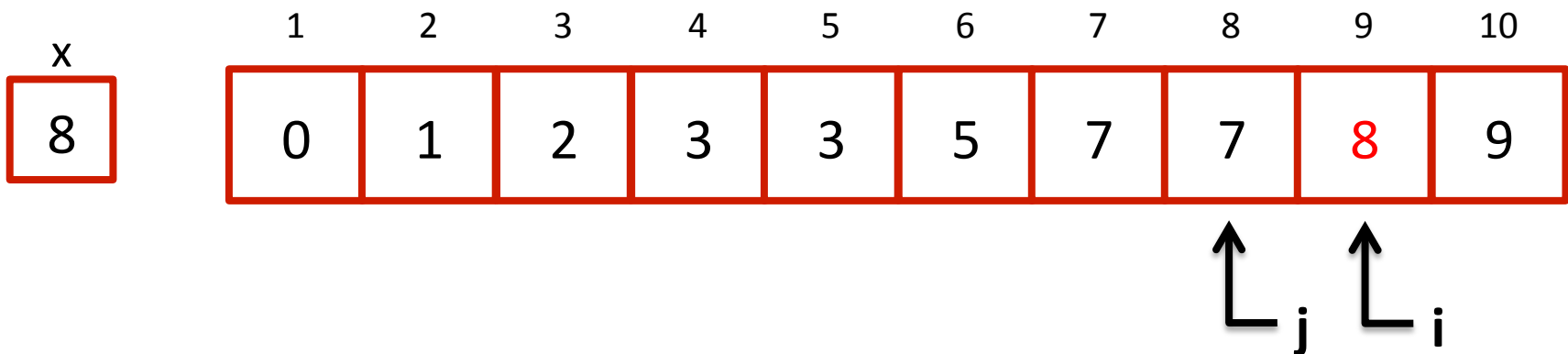
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



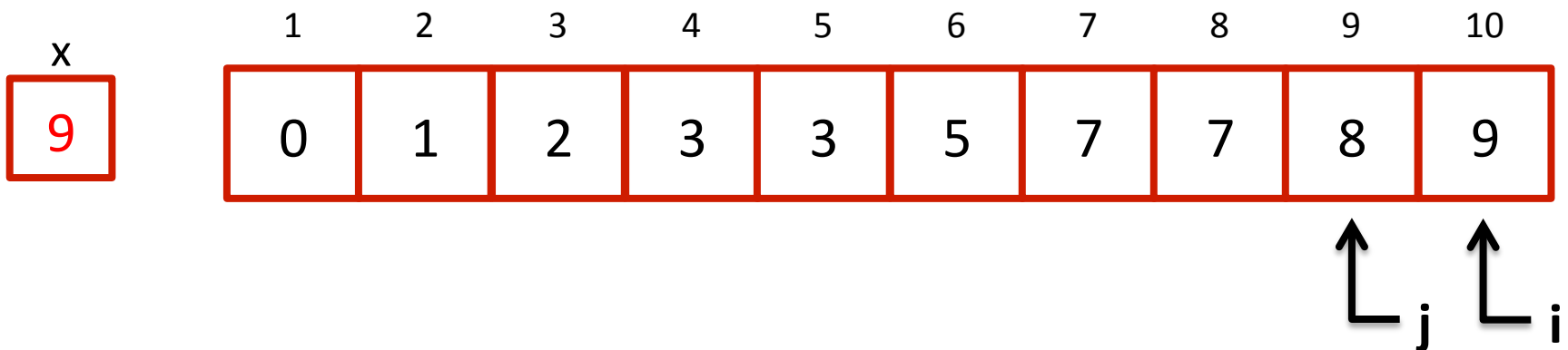
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



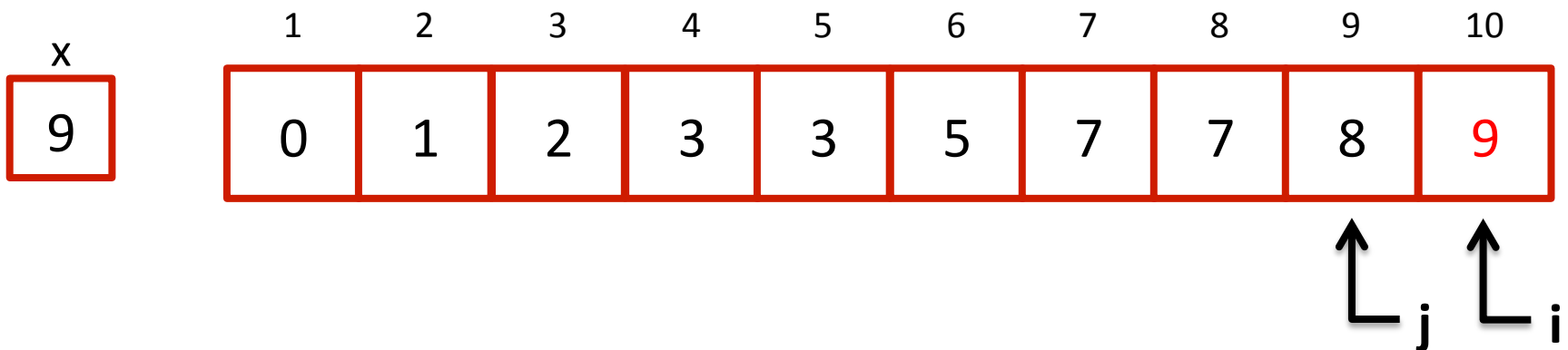
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```



Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])
```

```
  for i:=2 to n do
```

```
    x:=T[i];
```

```
    j:=i-1;
```

```
    while j>0 and T[j]>x do
```

```
      T[j+1]:=T[j];
```

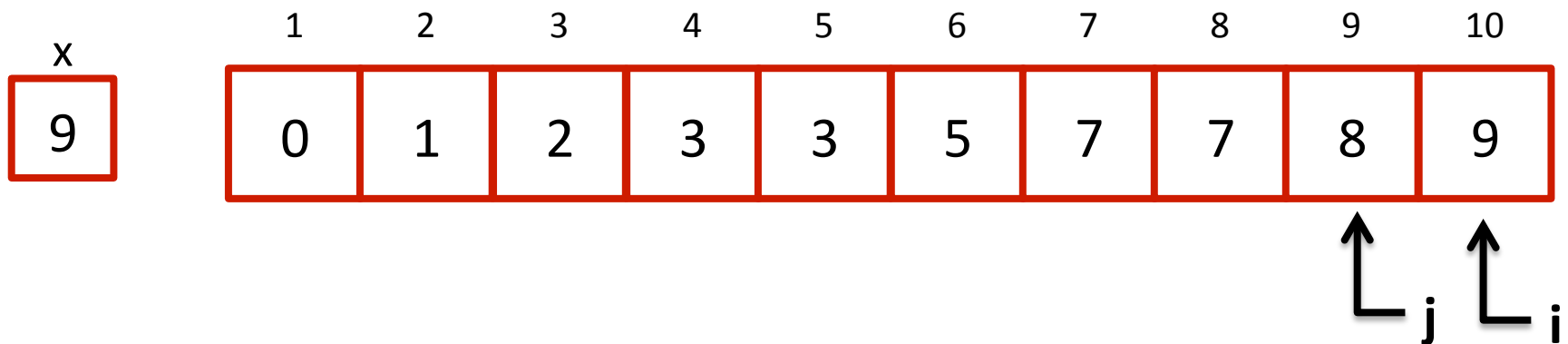
```
      j:=j-1
```

```
    end while;
```

```
    T[j+1]:=x
```

```
  end for
```

```
end procedure
```



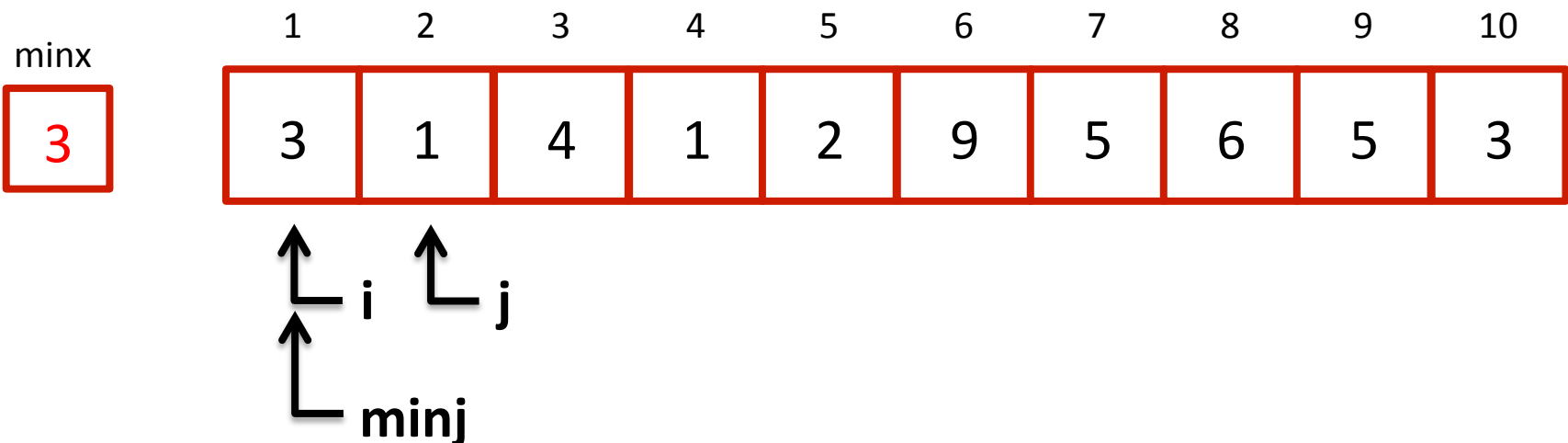
Insertion sort: Best case analysis

```
procedure Insertion sort (var T[1..n])  
  for i:=2 to n do  
    x:=T[i];  
    j:=i-1;  
    while j>0 and T[j]>x do  
      T[j+1]:=T[j];  
      j:=j-1  
    end while;  
    T[j+1]:=x  
  end for  
end procedure
```

1	2	3	4	5	6	7	8	9	10
0	1	2	3	3	5	7	7	8	9

Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

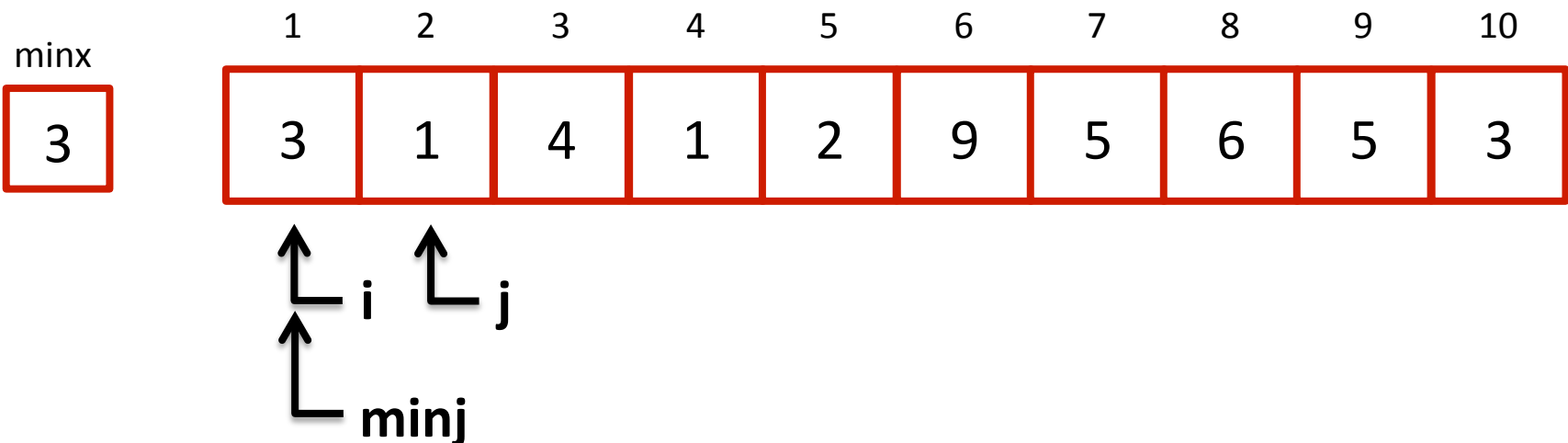
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

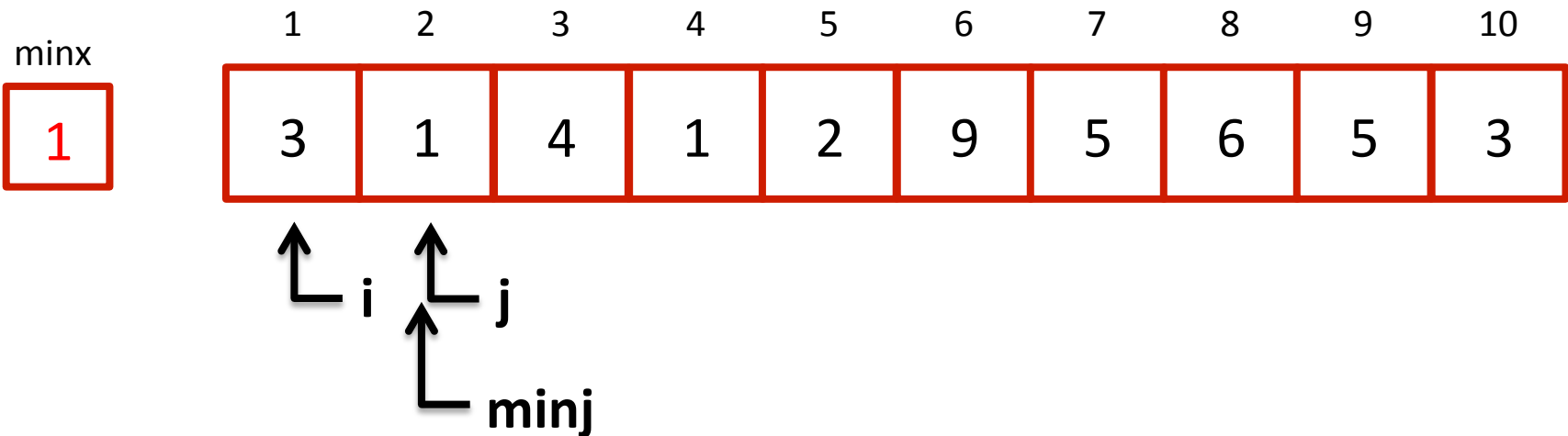
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

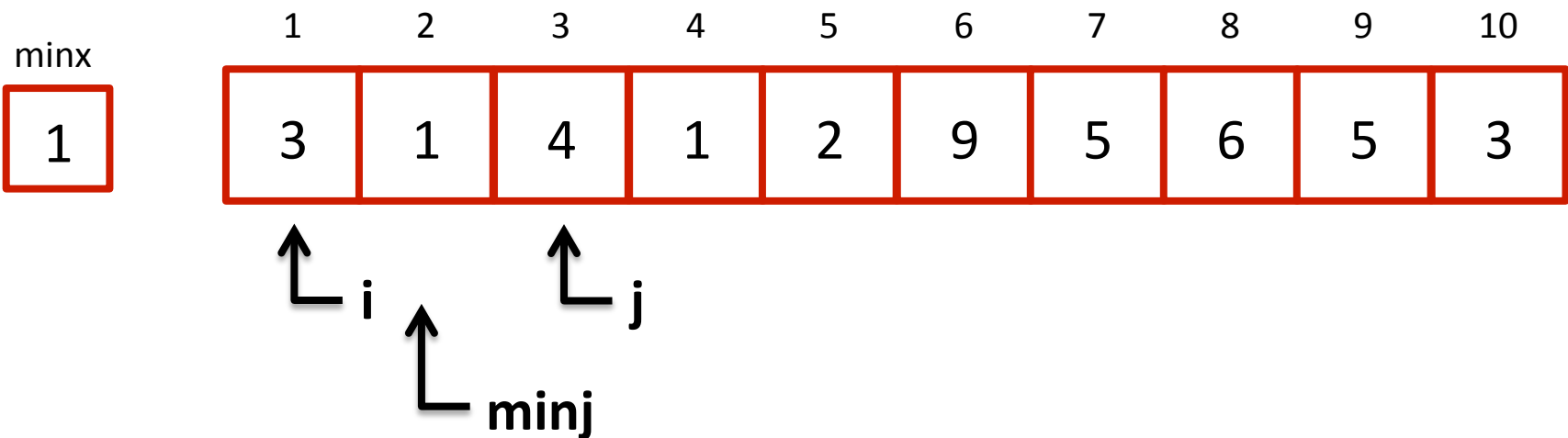
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

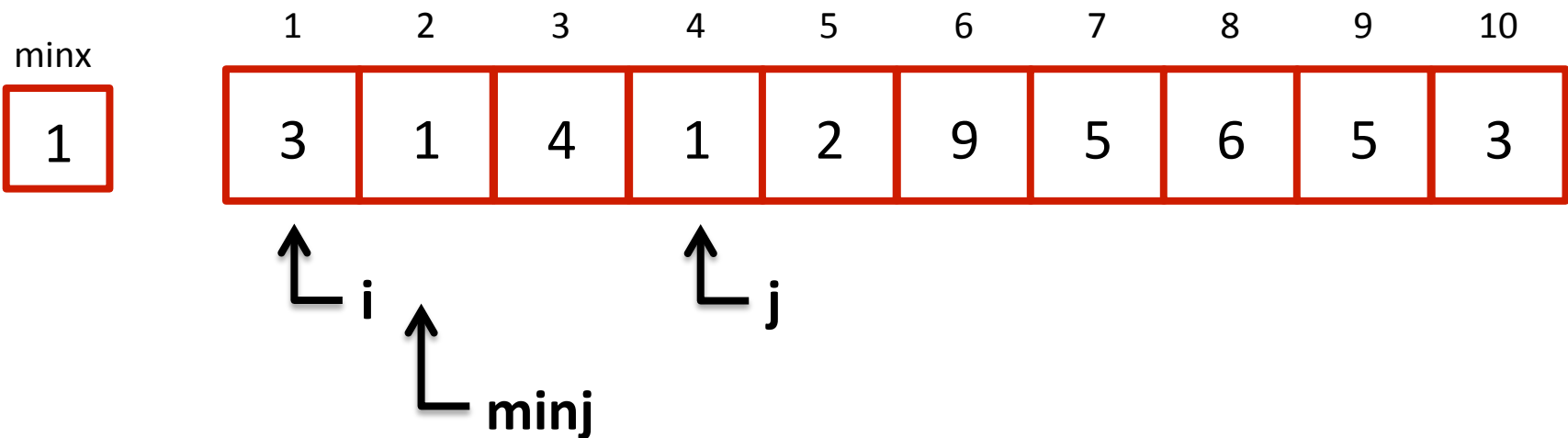
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

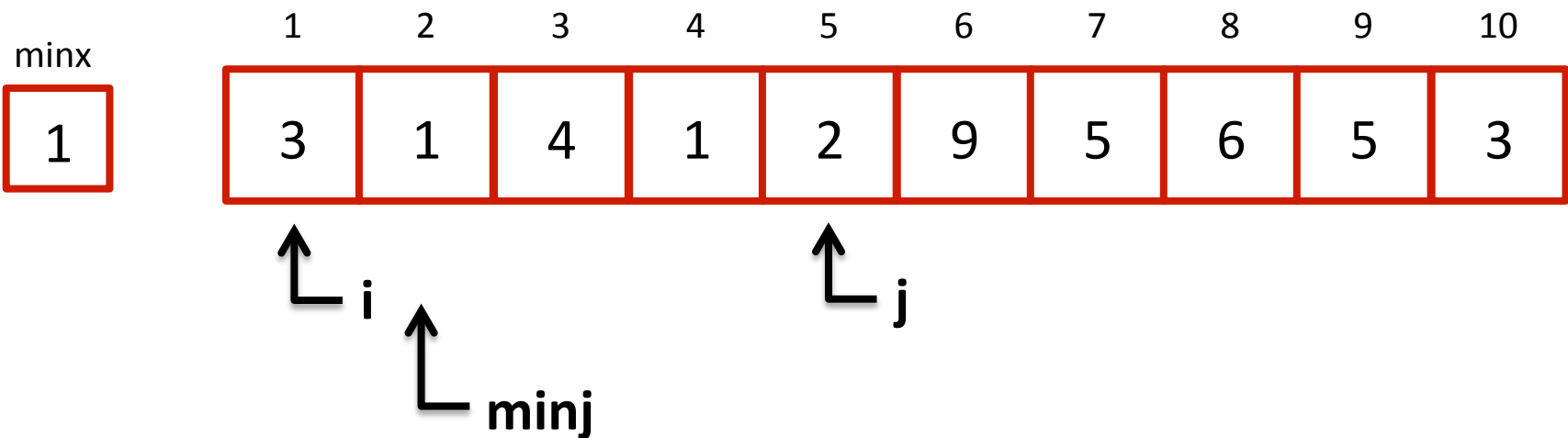
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

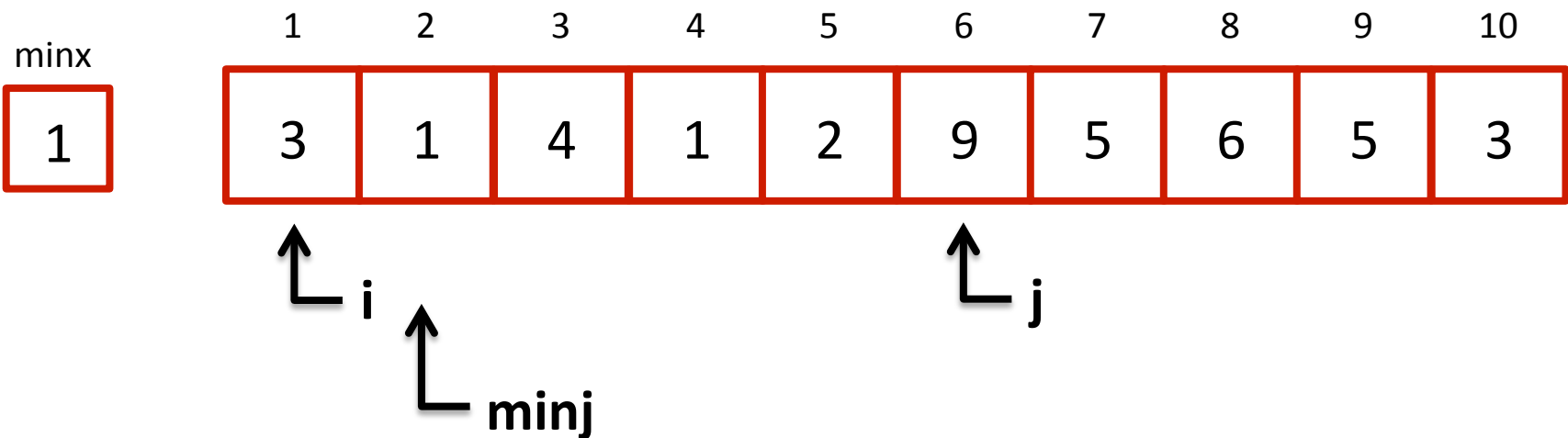
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

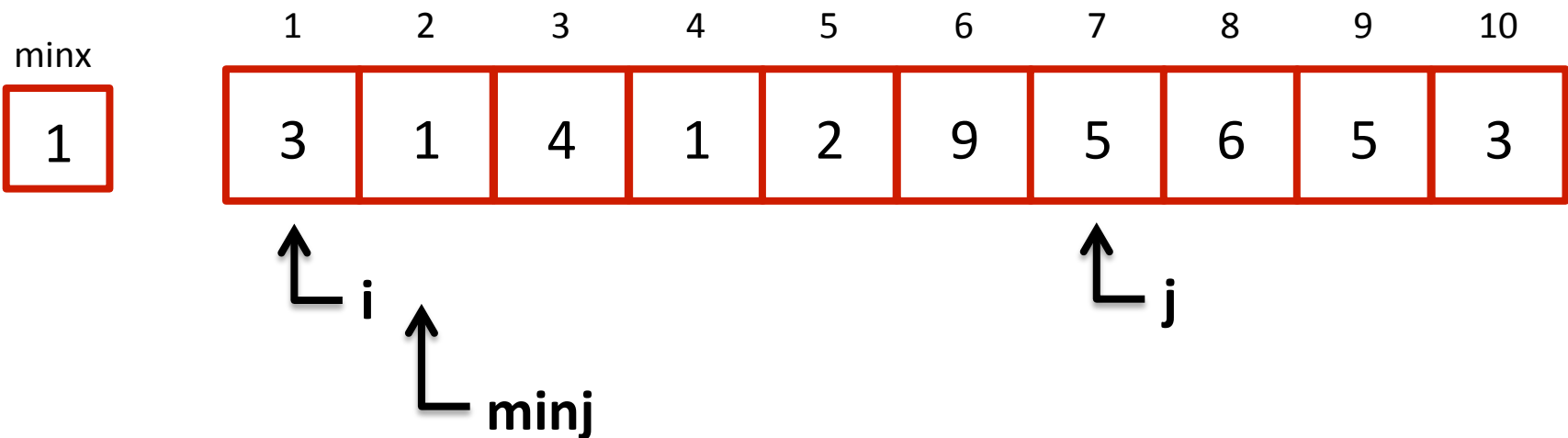
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

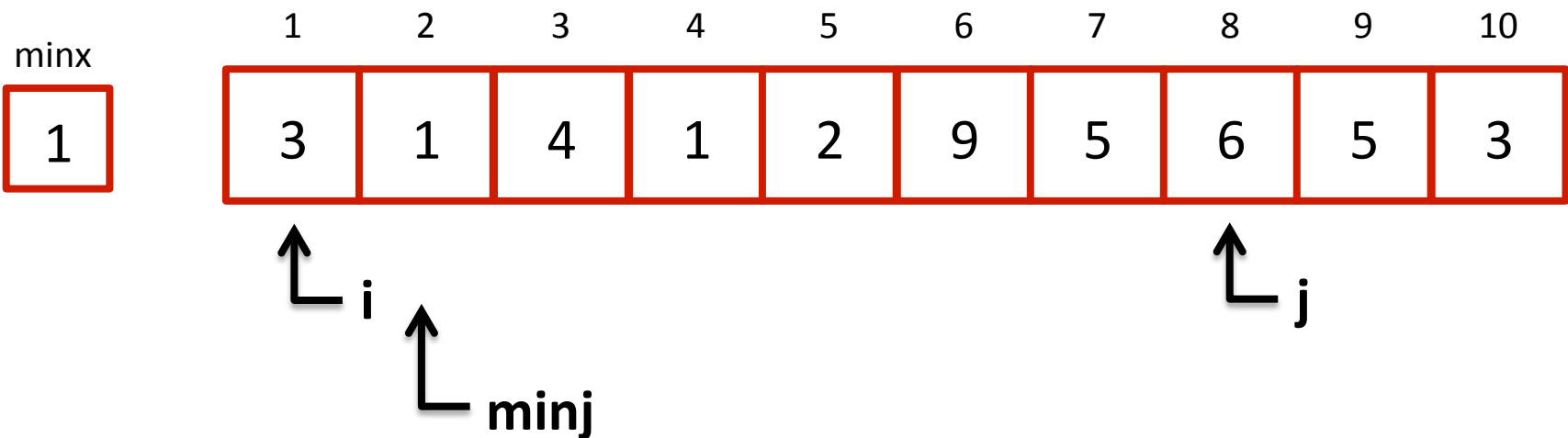
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

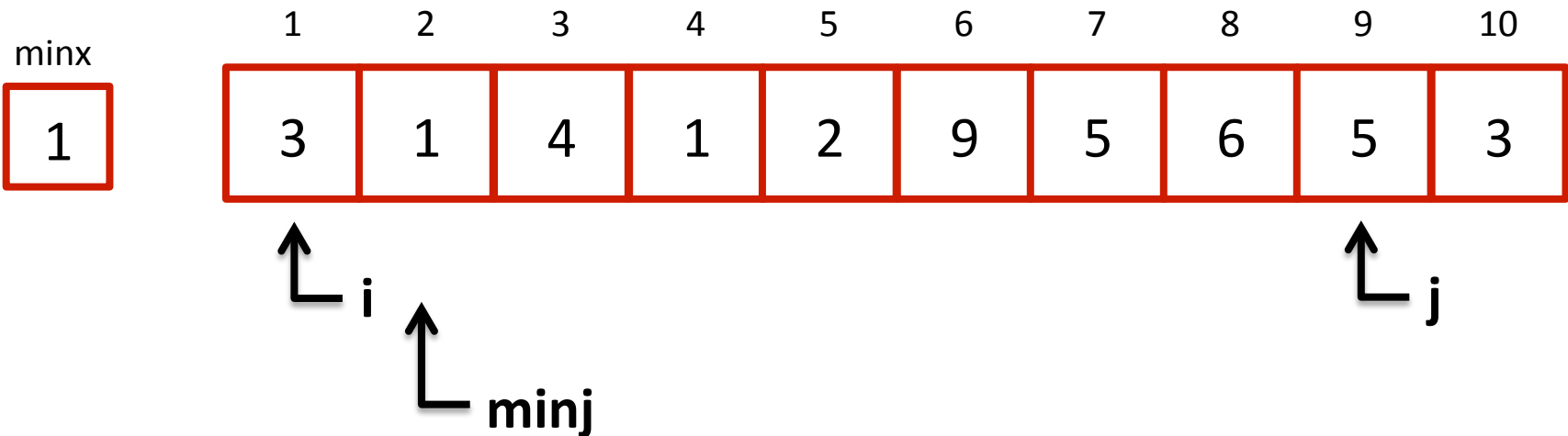
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

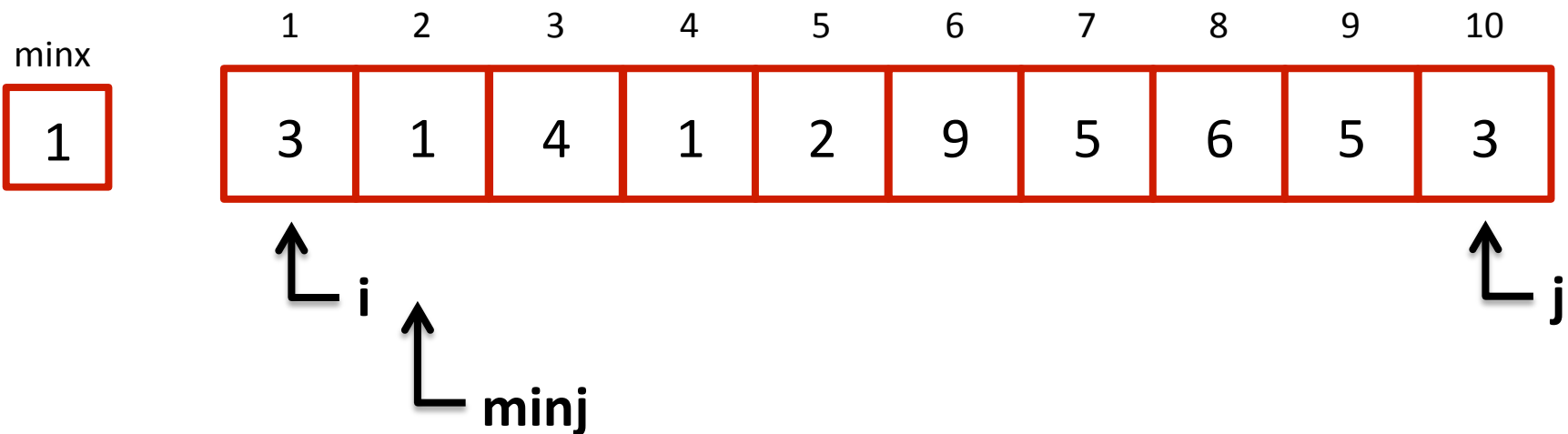
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

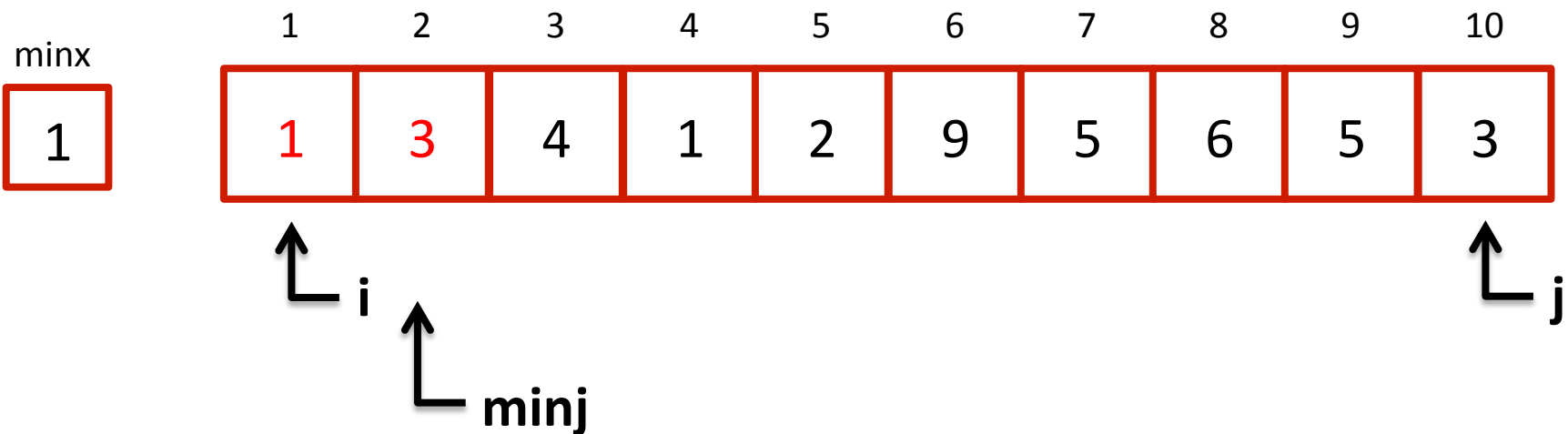
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

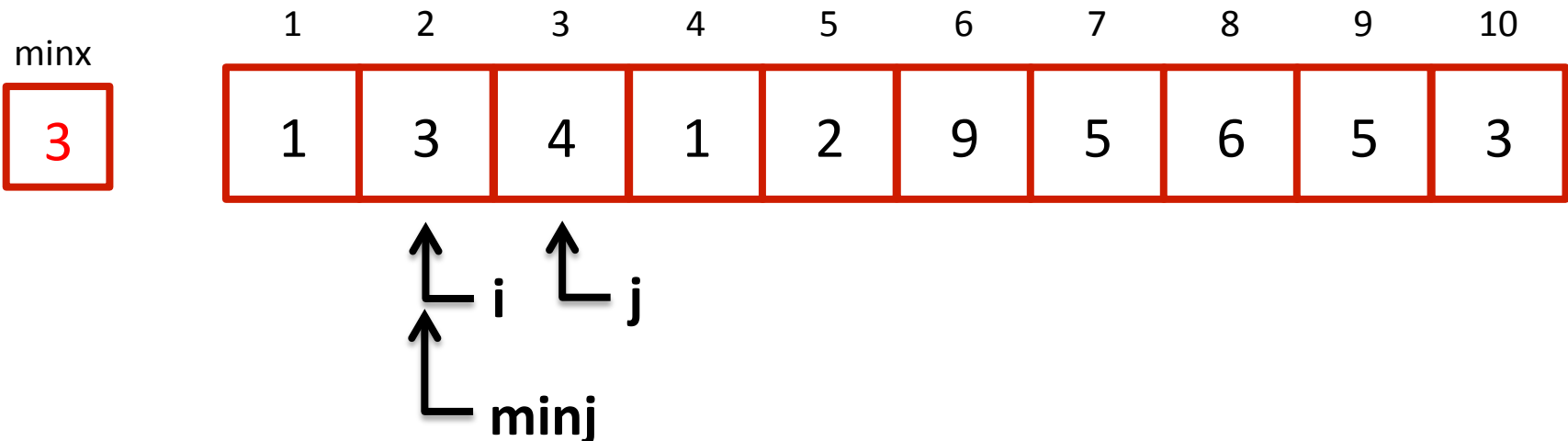
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

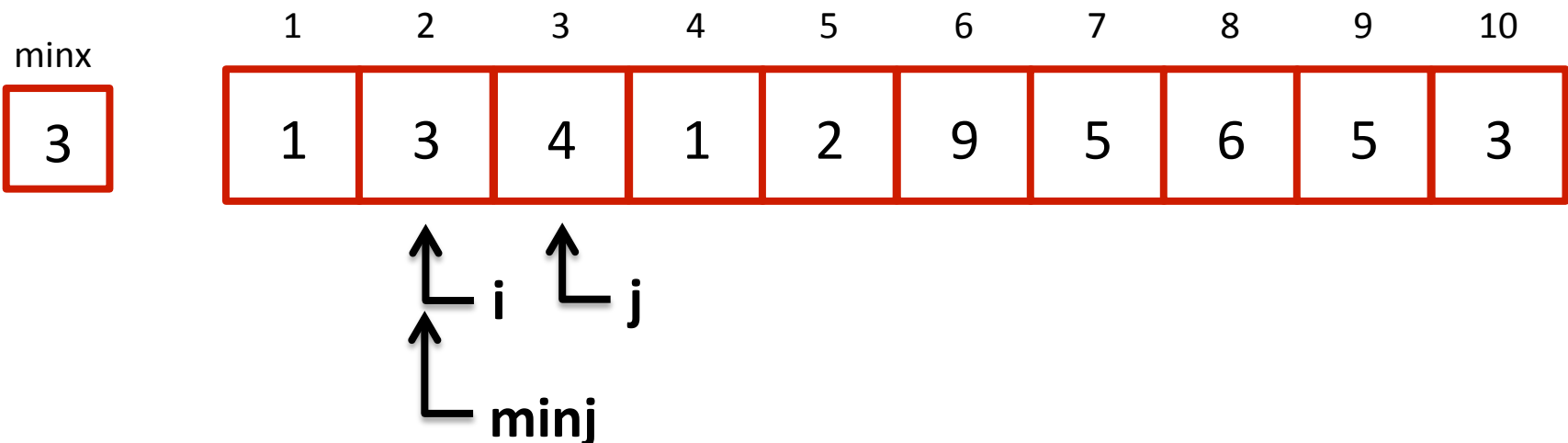
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

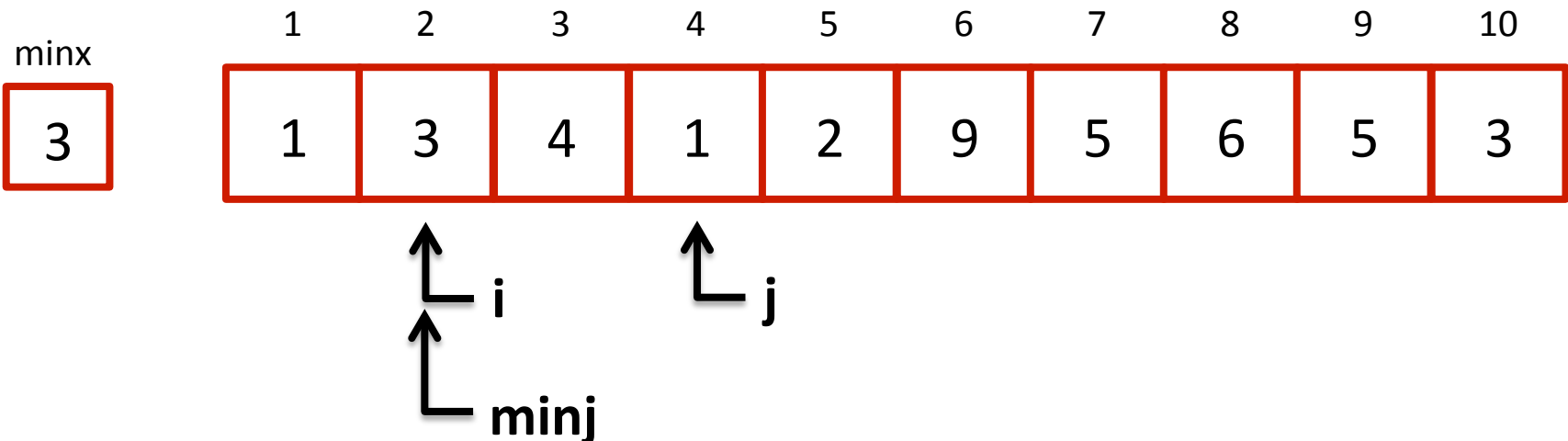
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

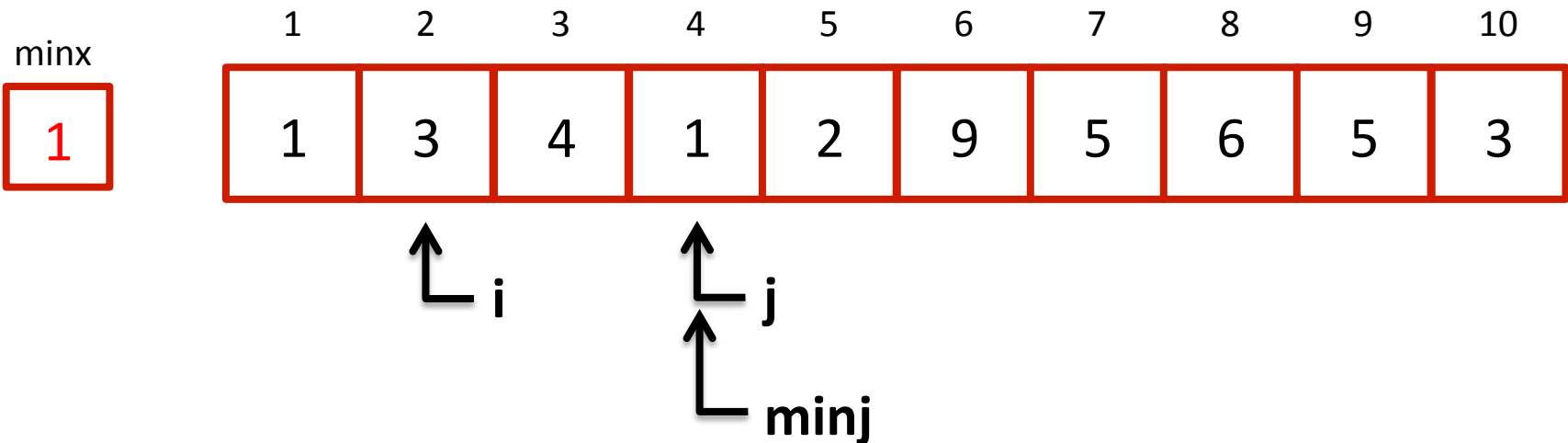
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

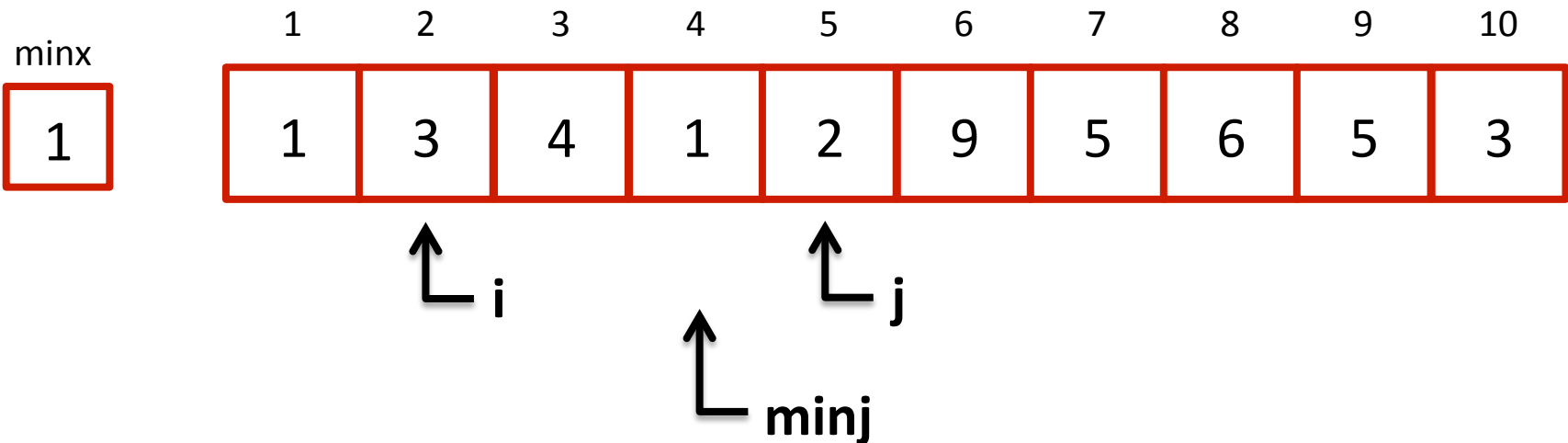
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

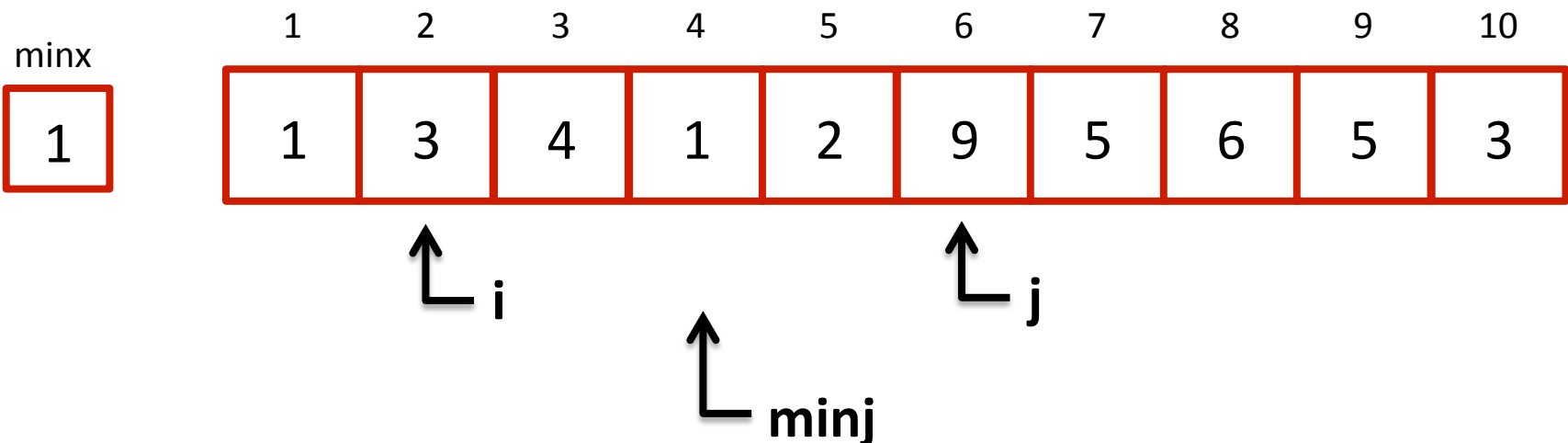
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

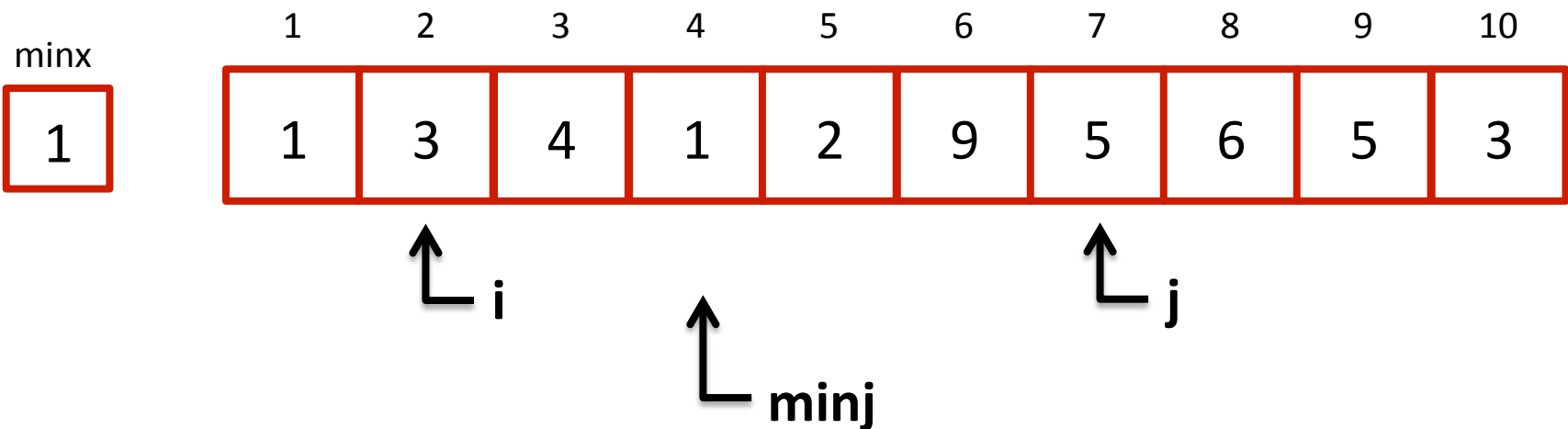
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

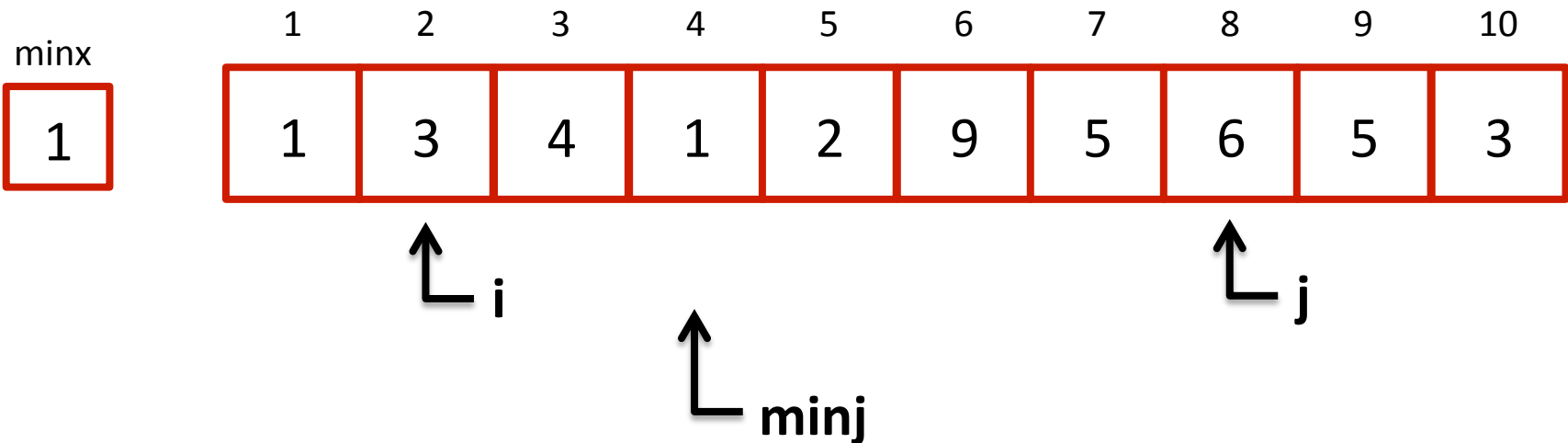
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

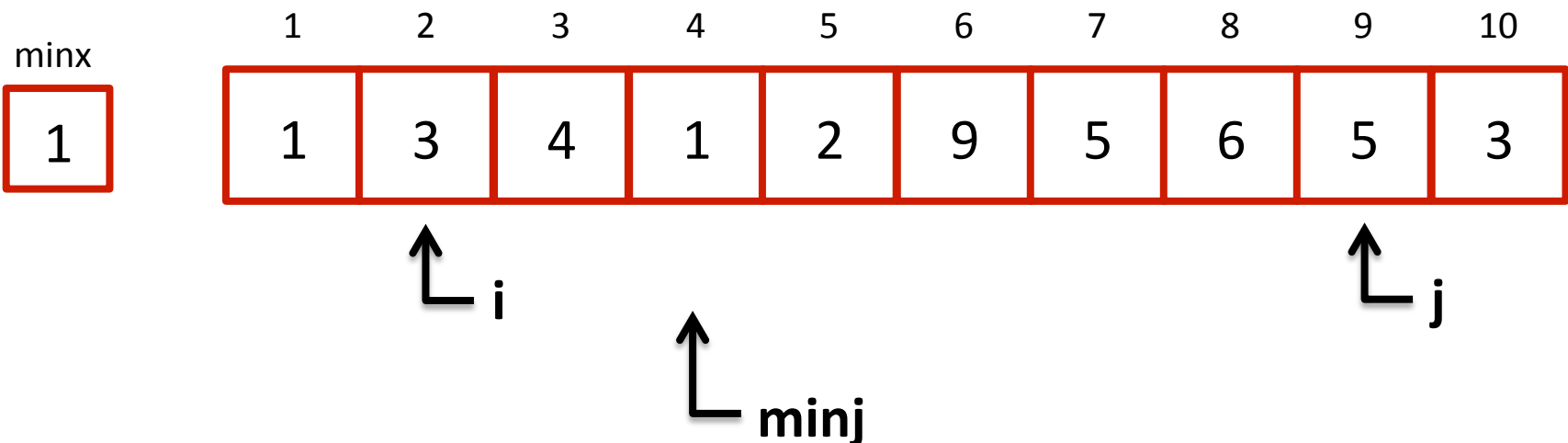
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

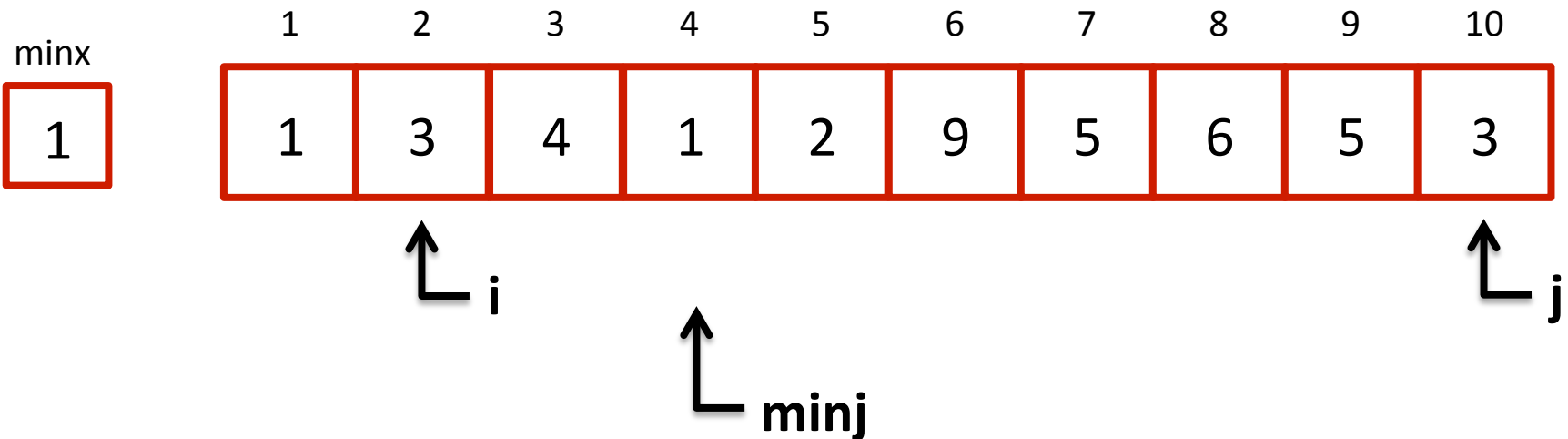
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

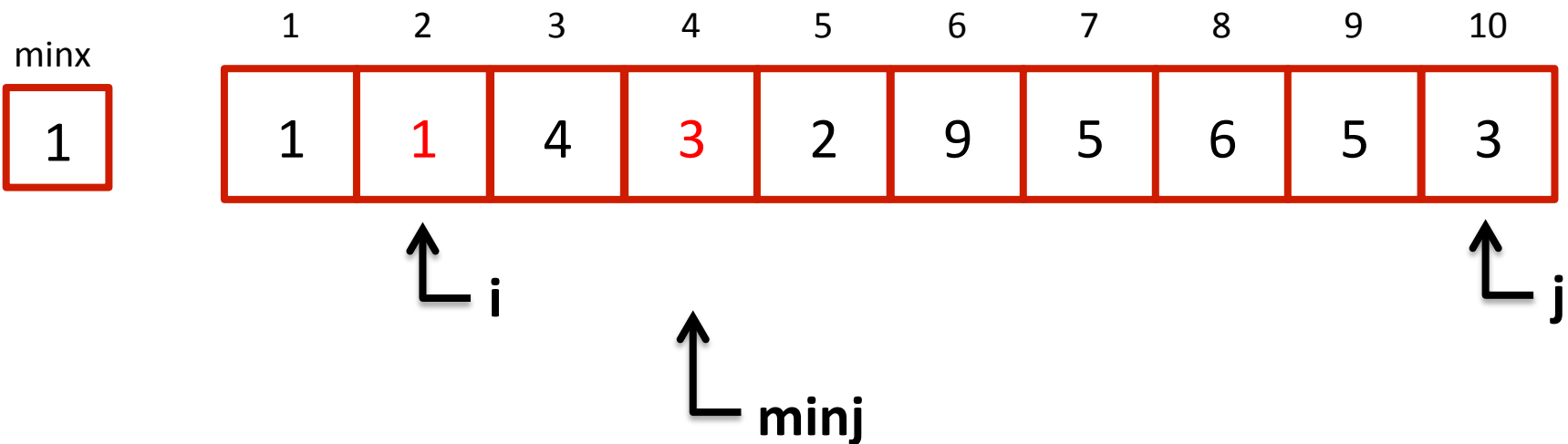
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

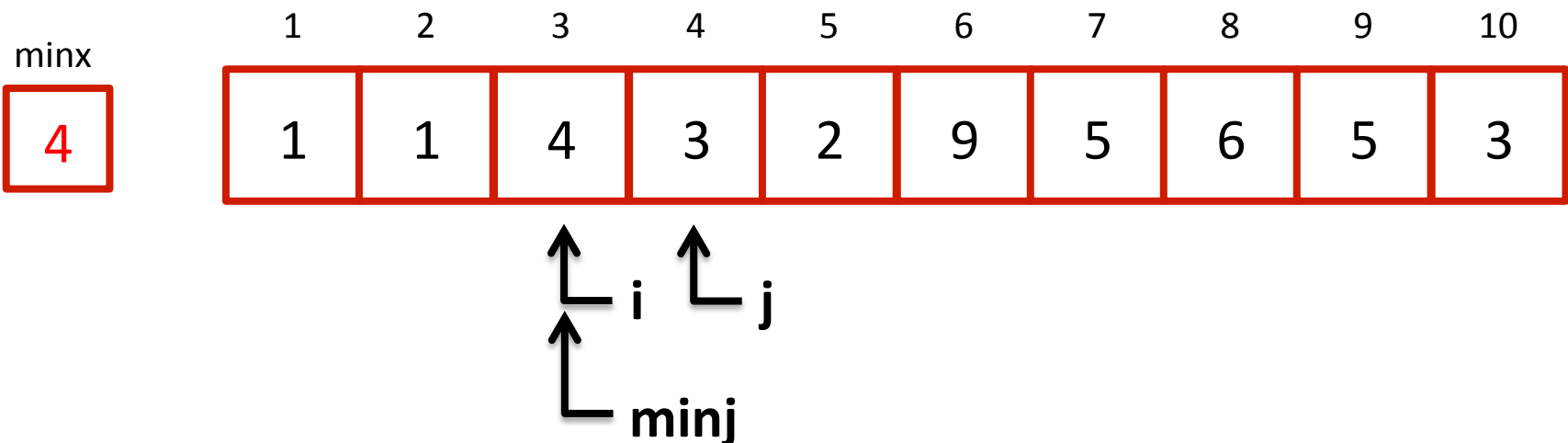
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

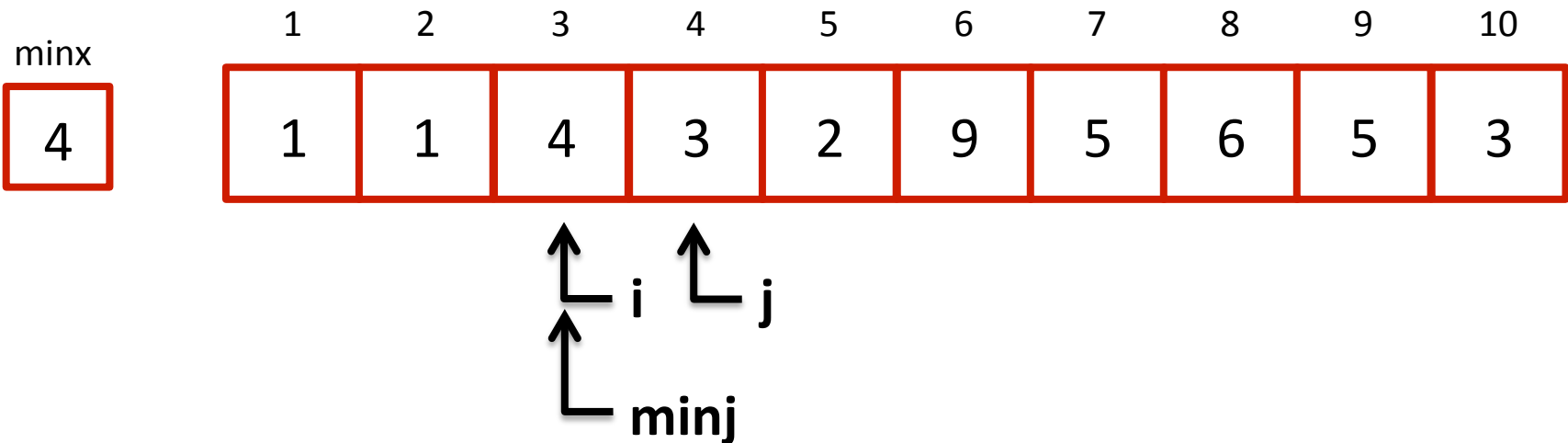
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

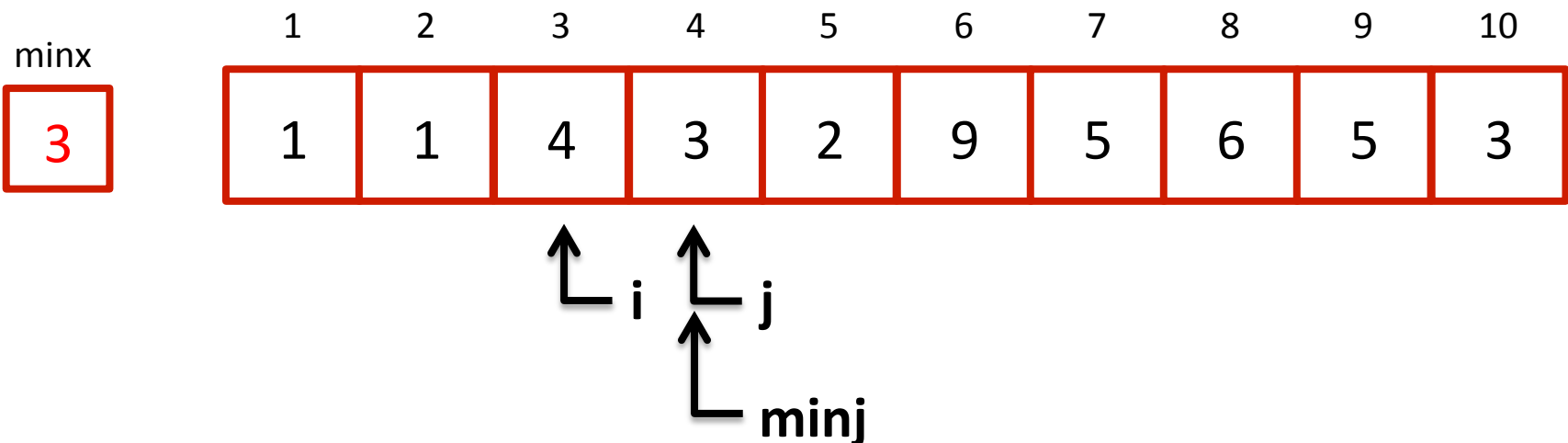
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

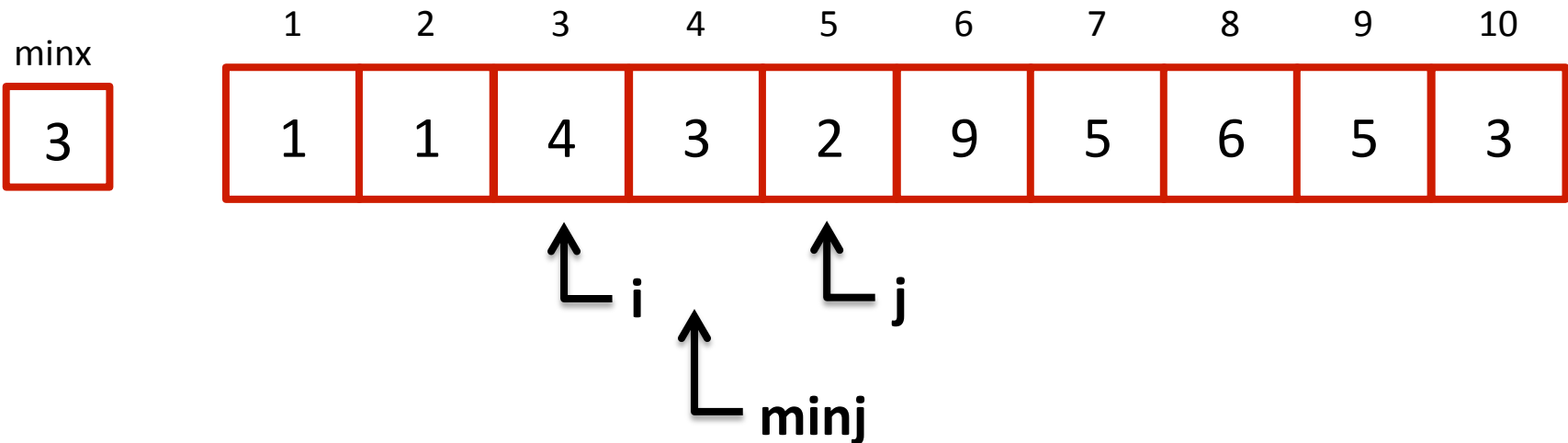
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

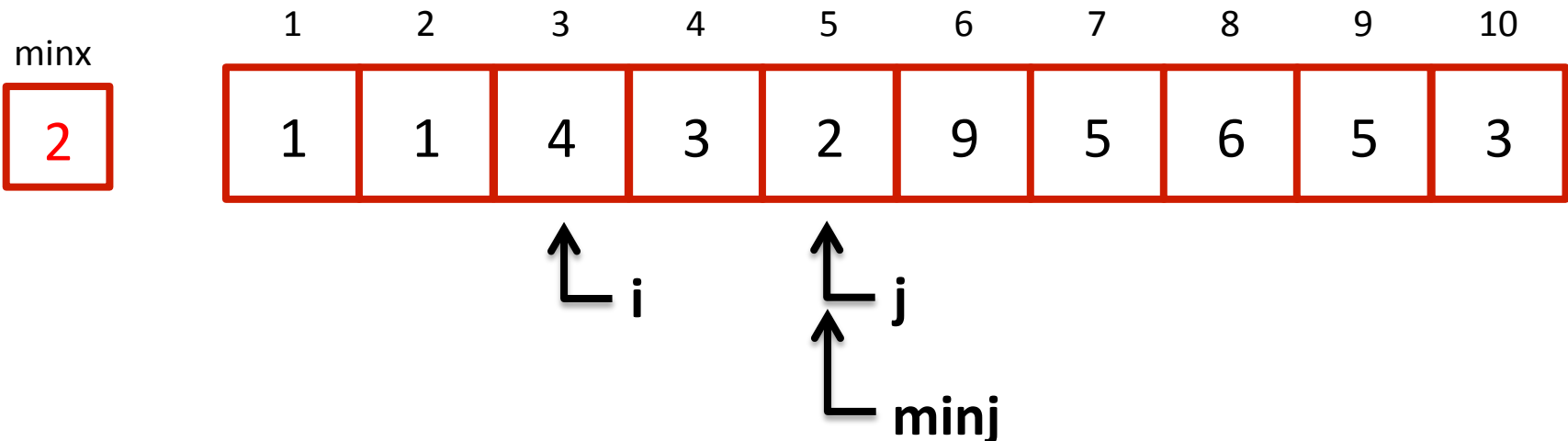
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

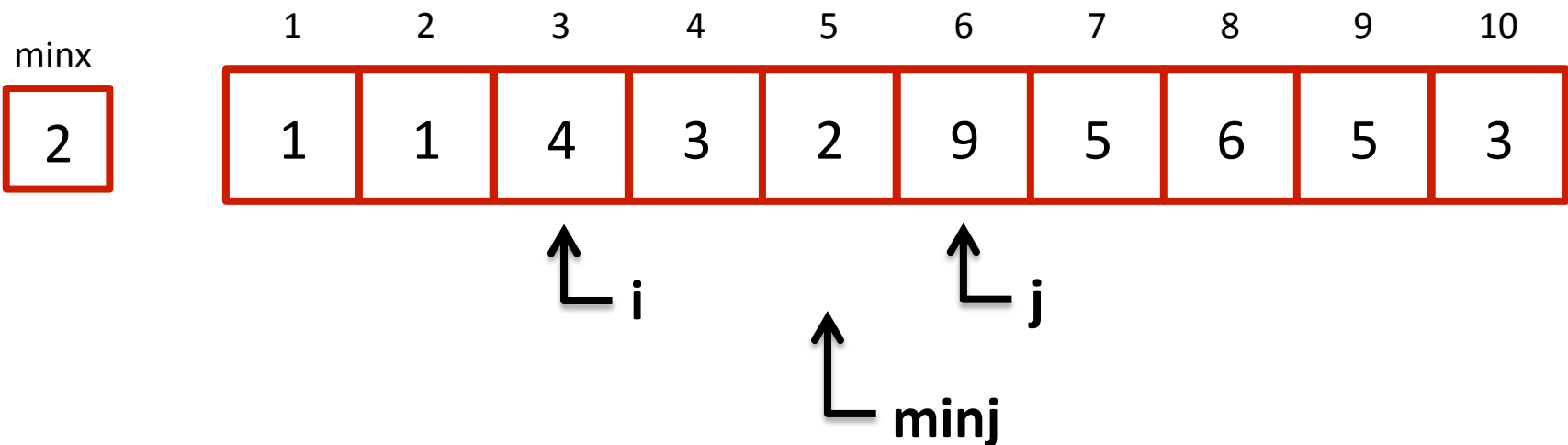
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

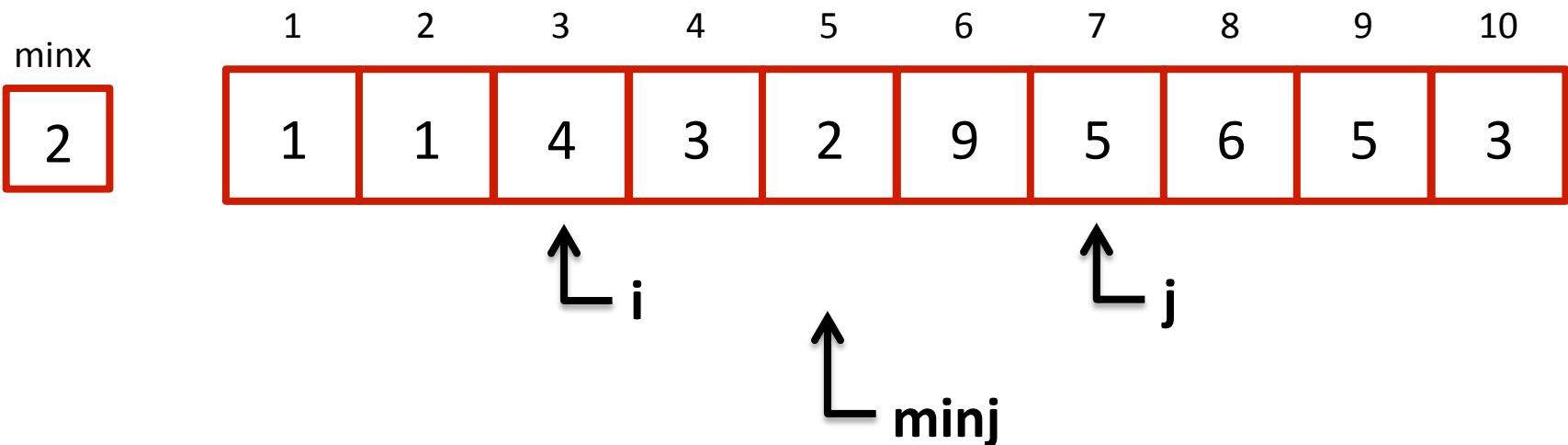
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

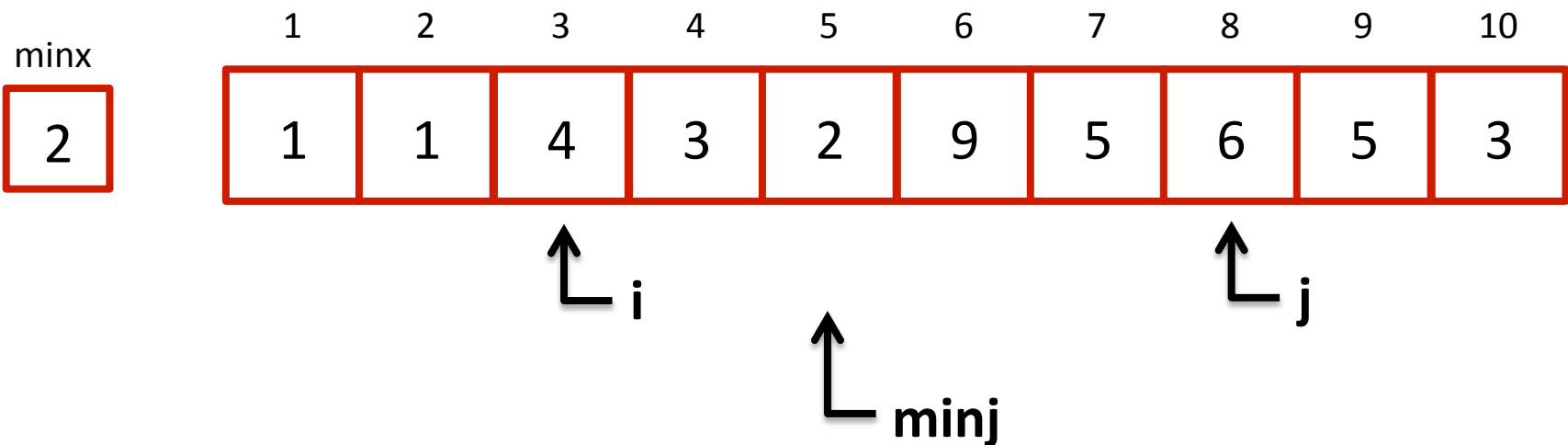
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

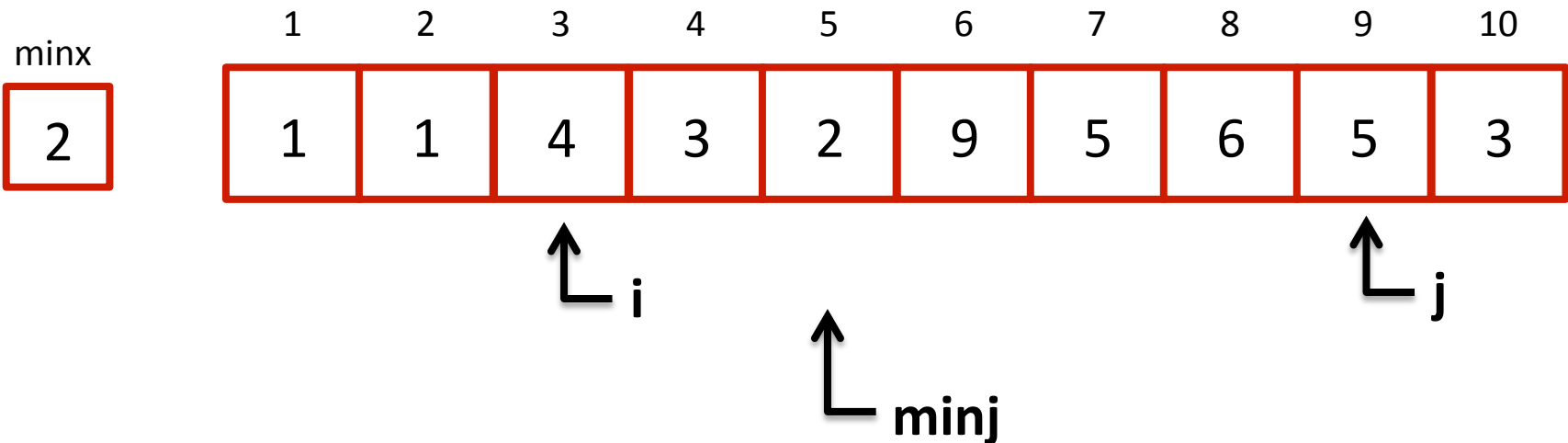
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

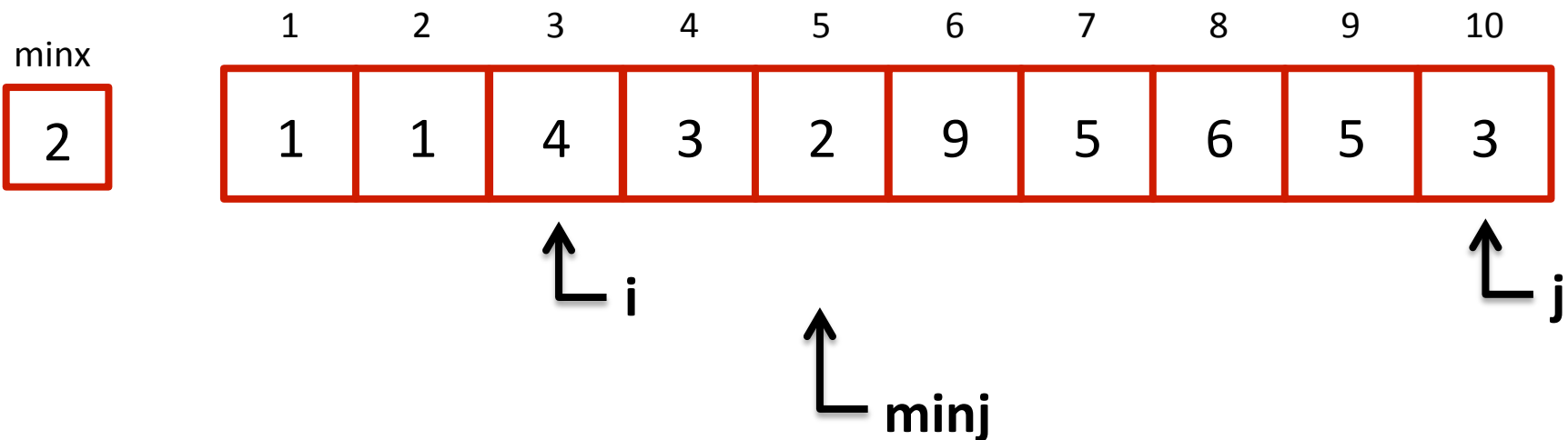
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

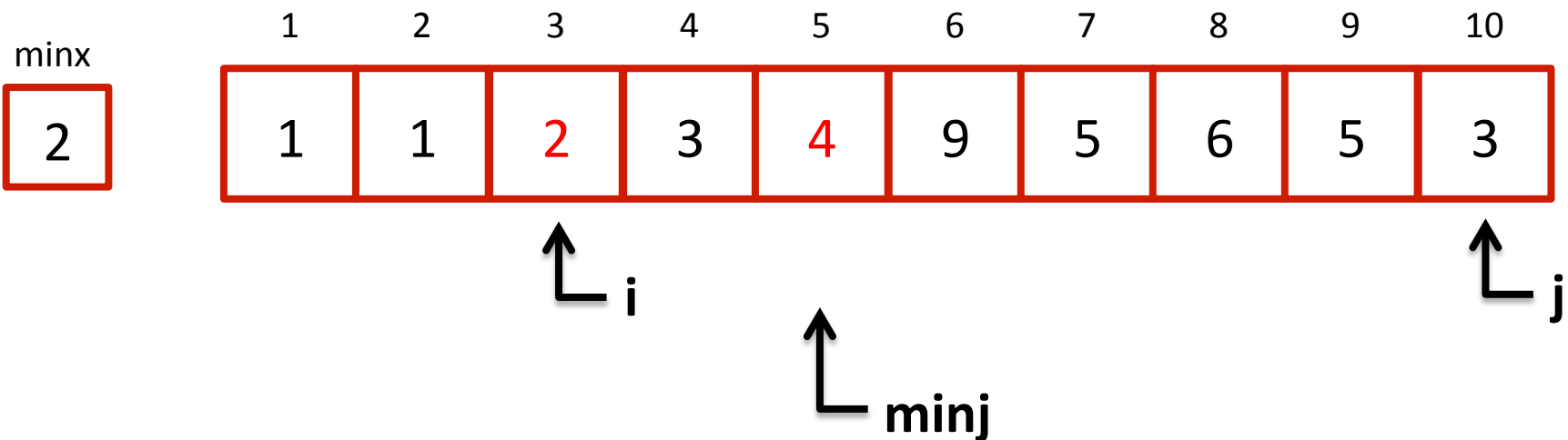
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

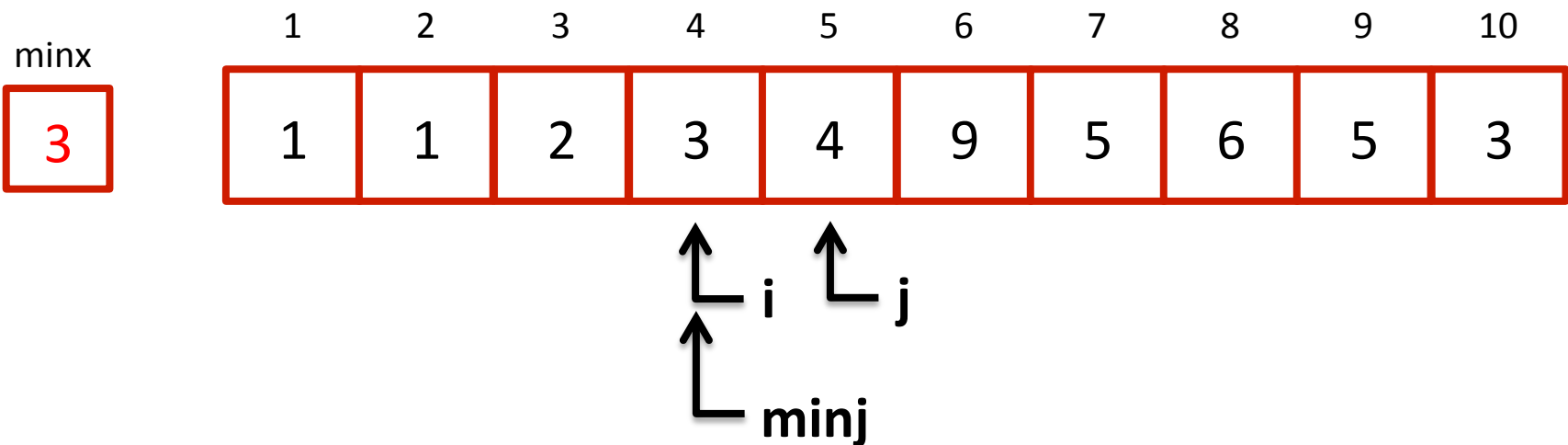
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

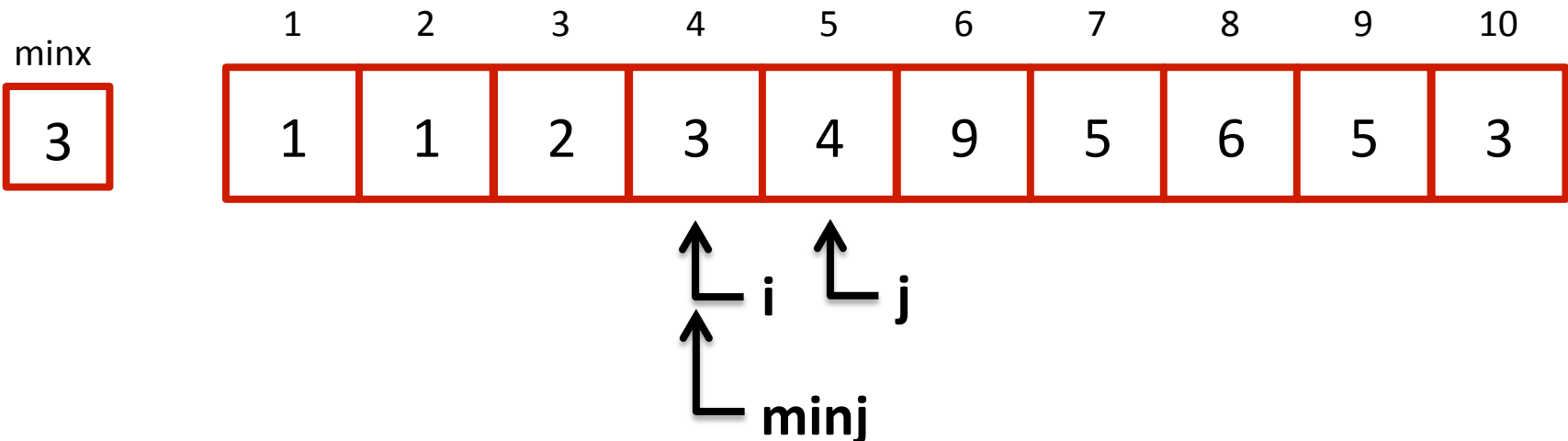
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

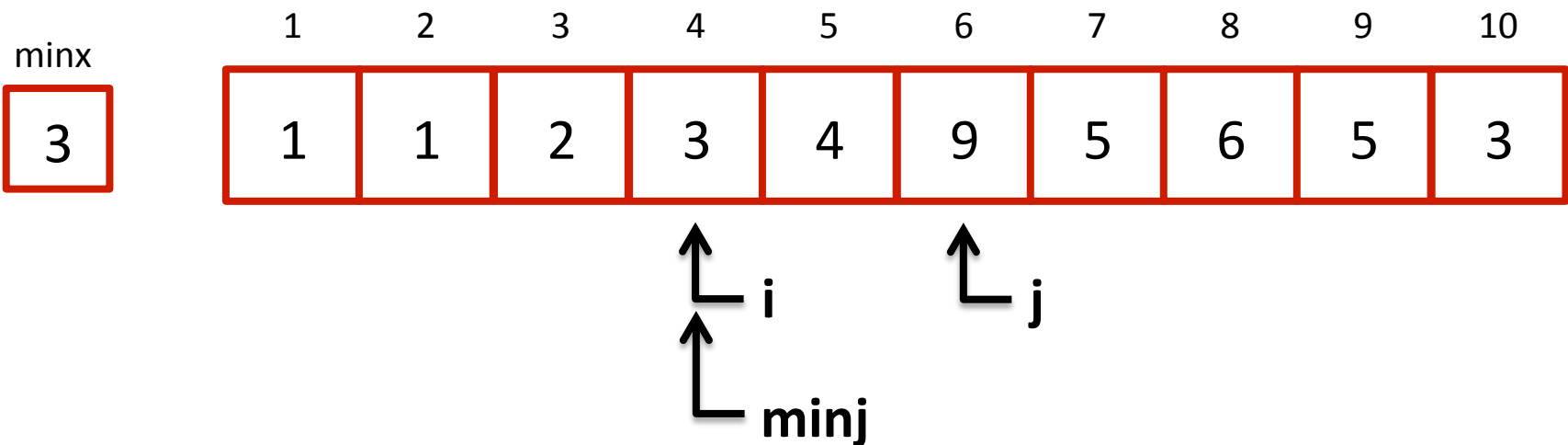
```
  end for
```

```
end procedure
```



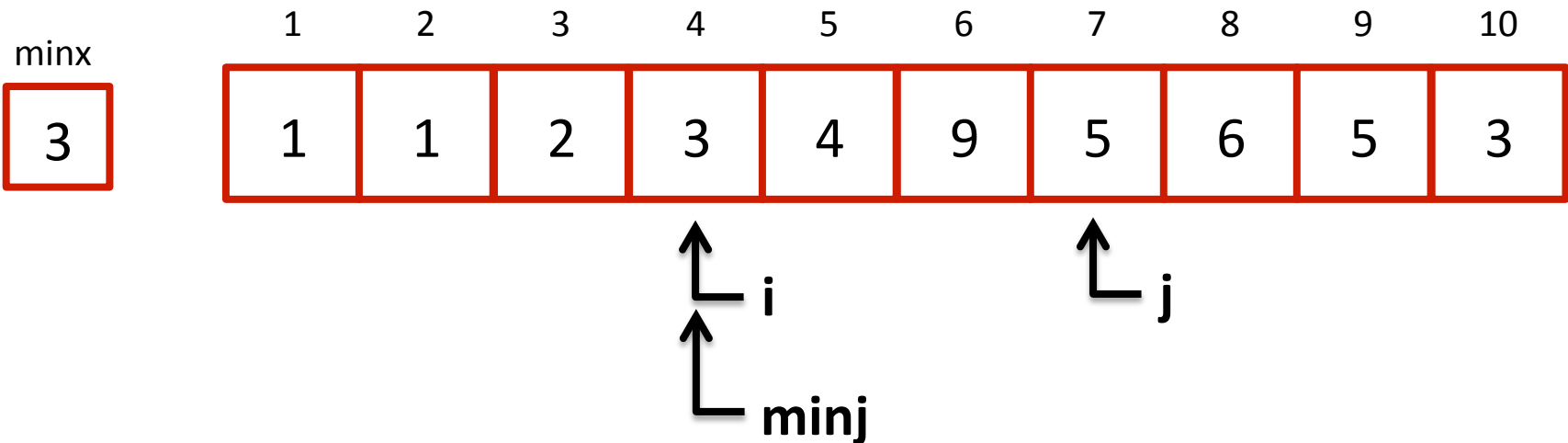
Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

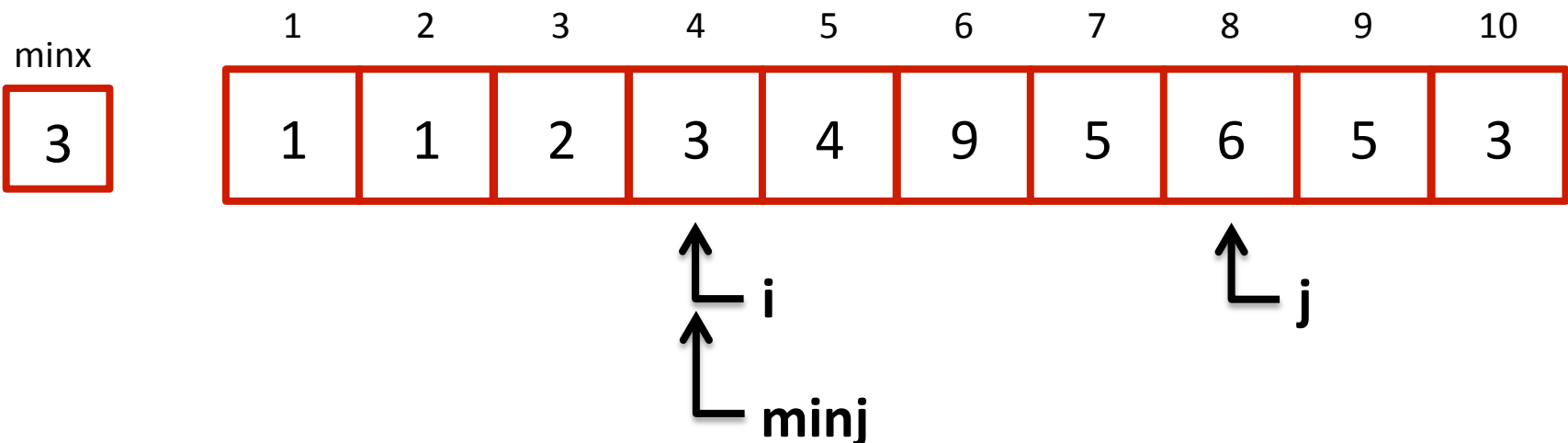
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

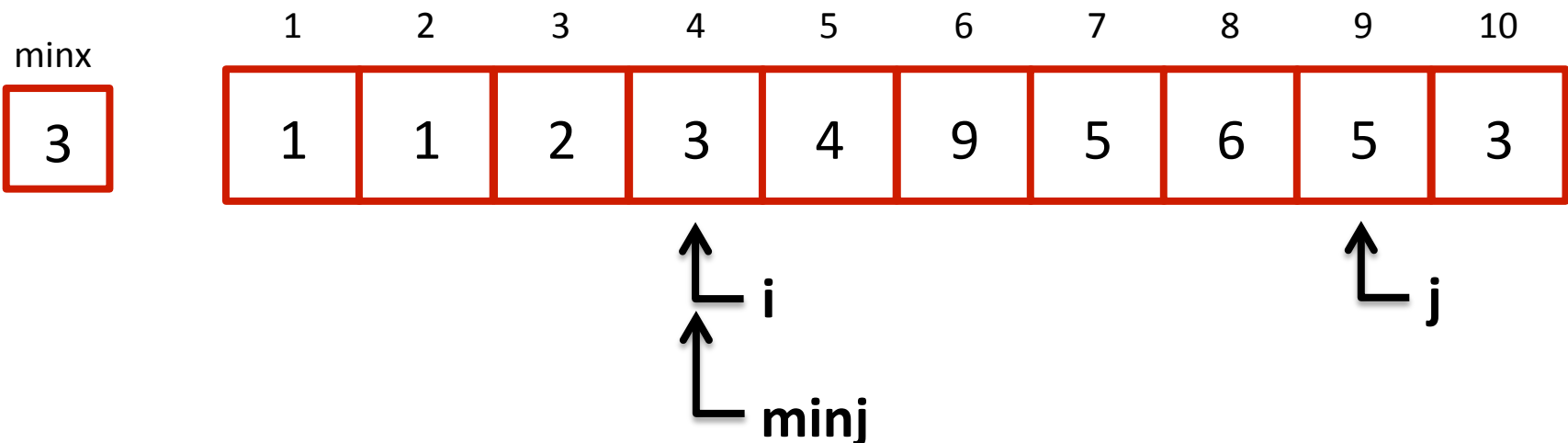
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

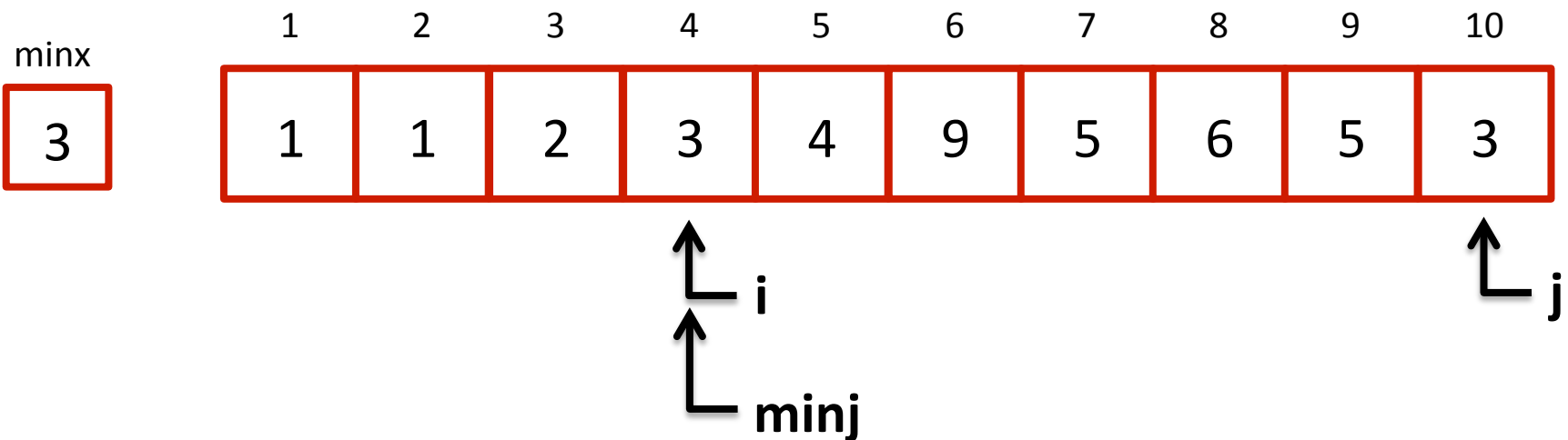
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

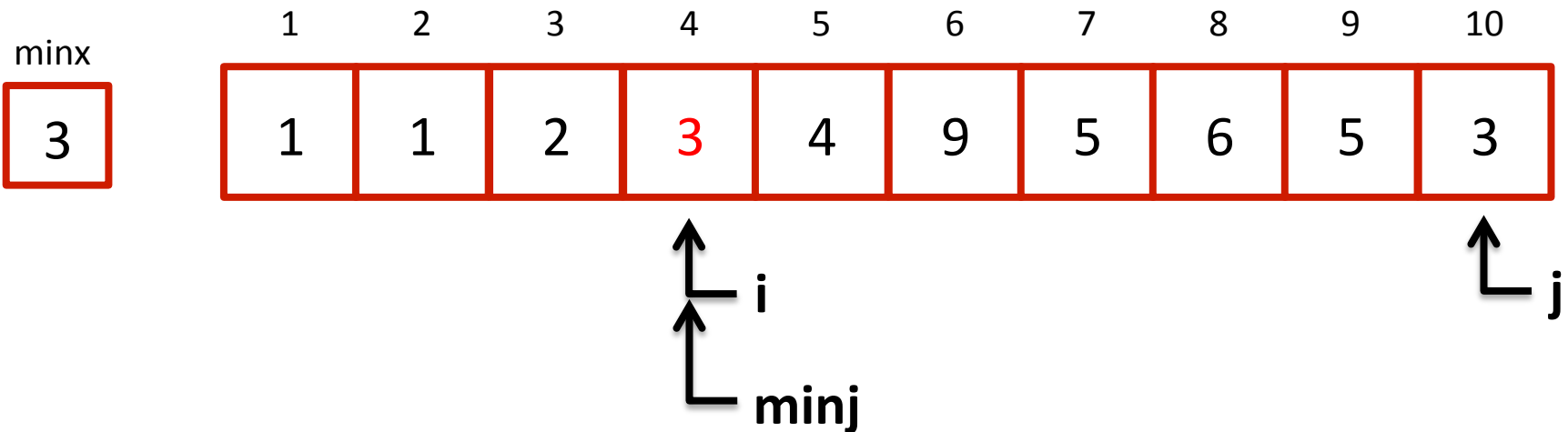
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

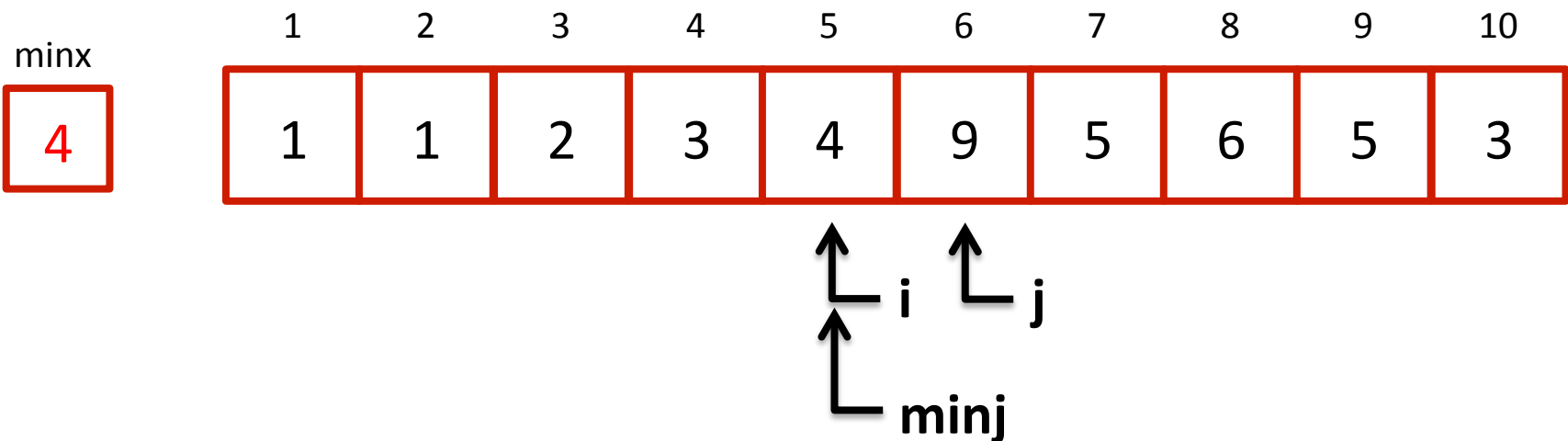
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

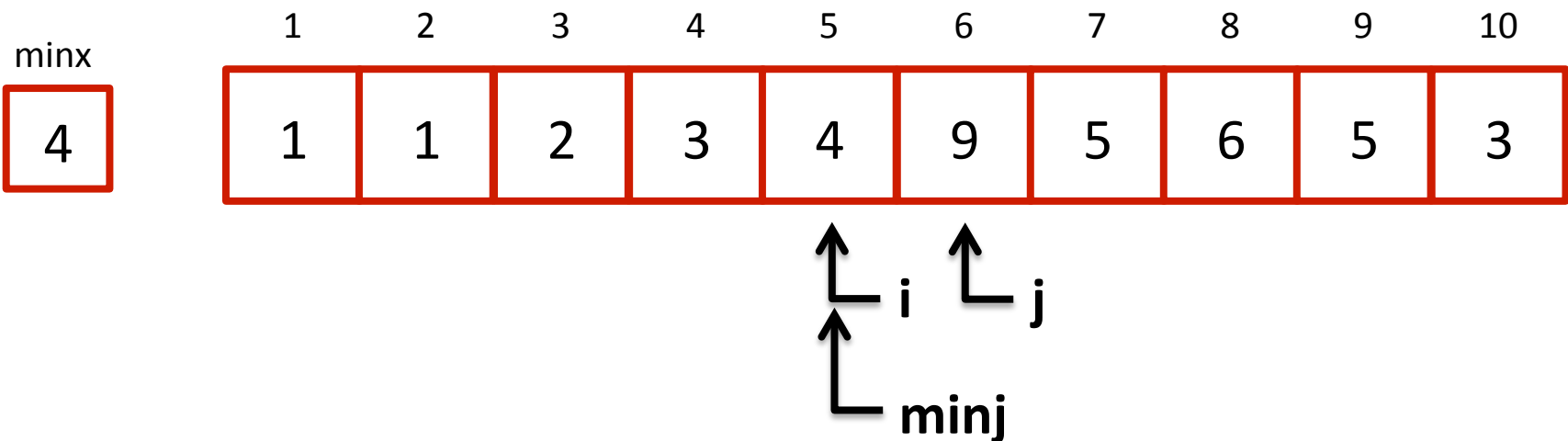
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

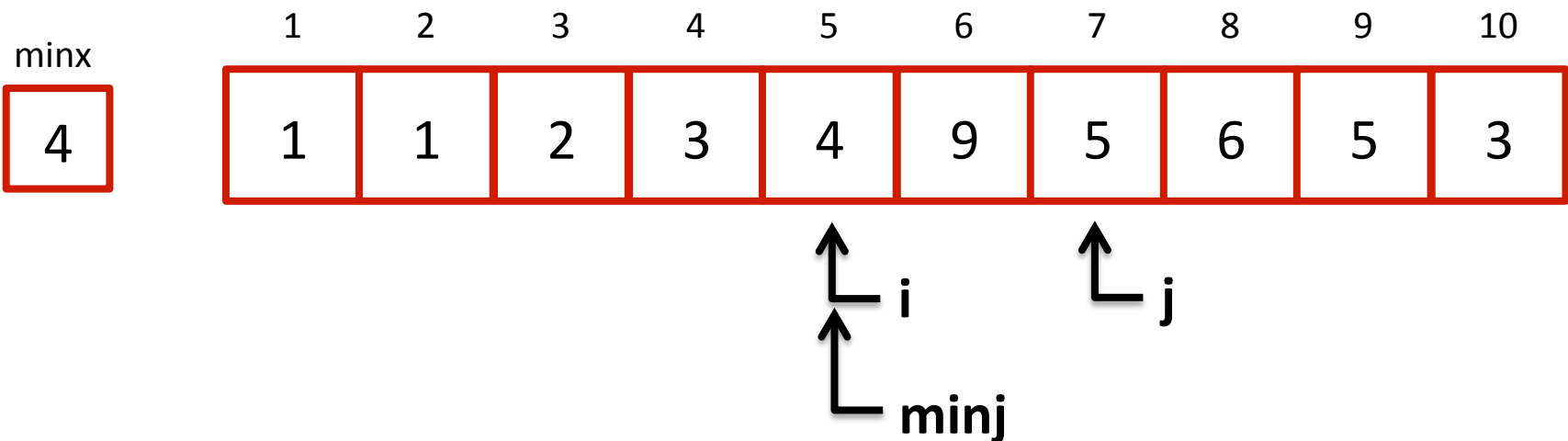
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

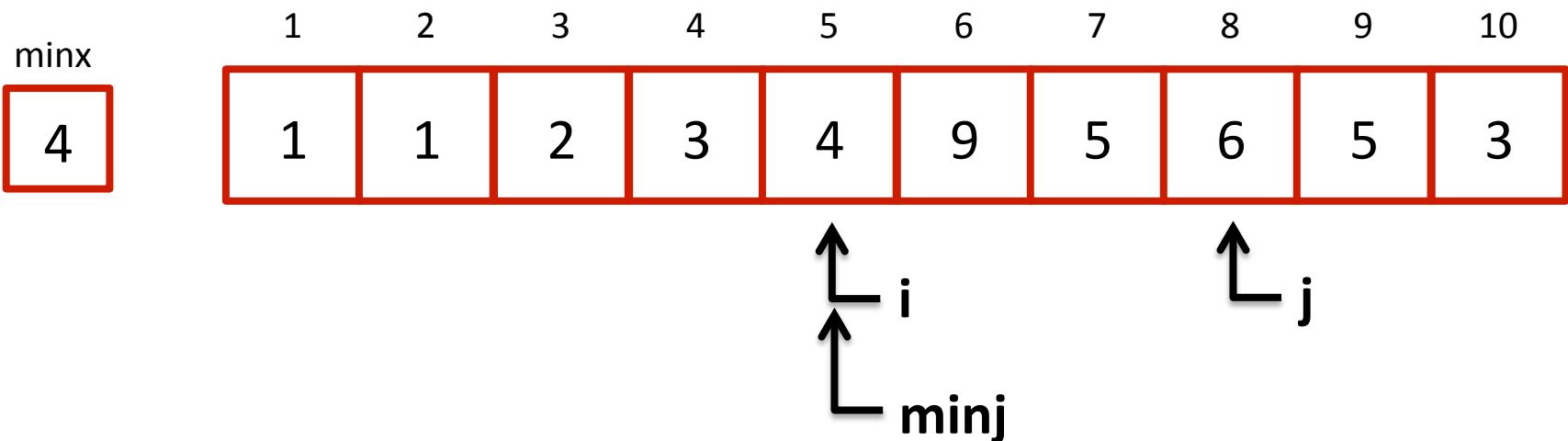
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

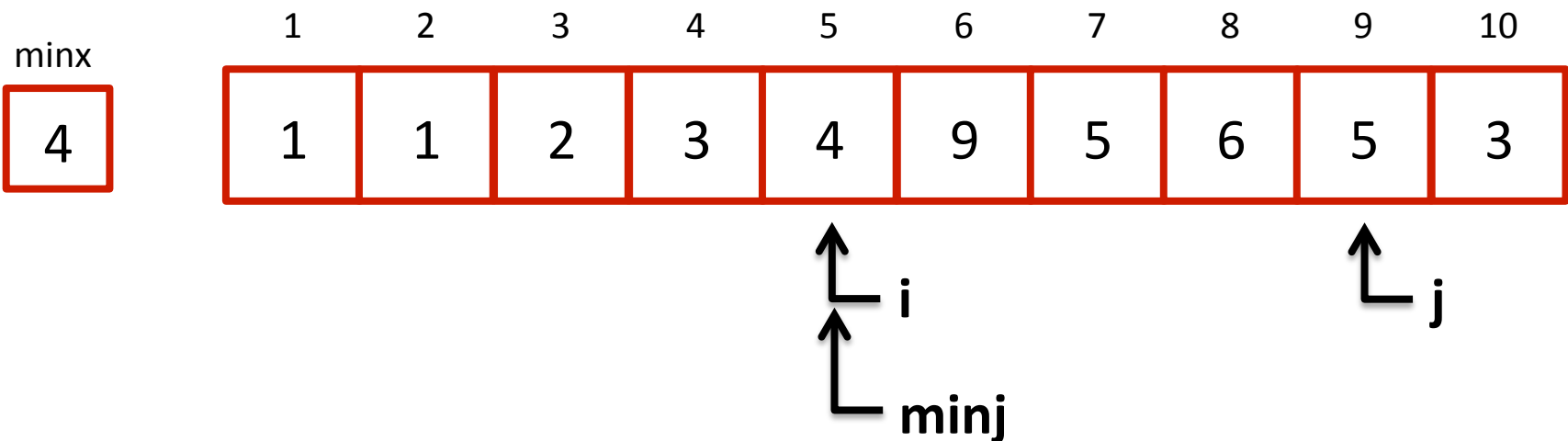
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

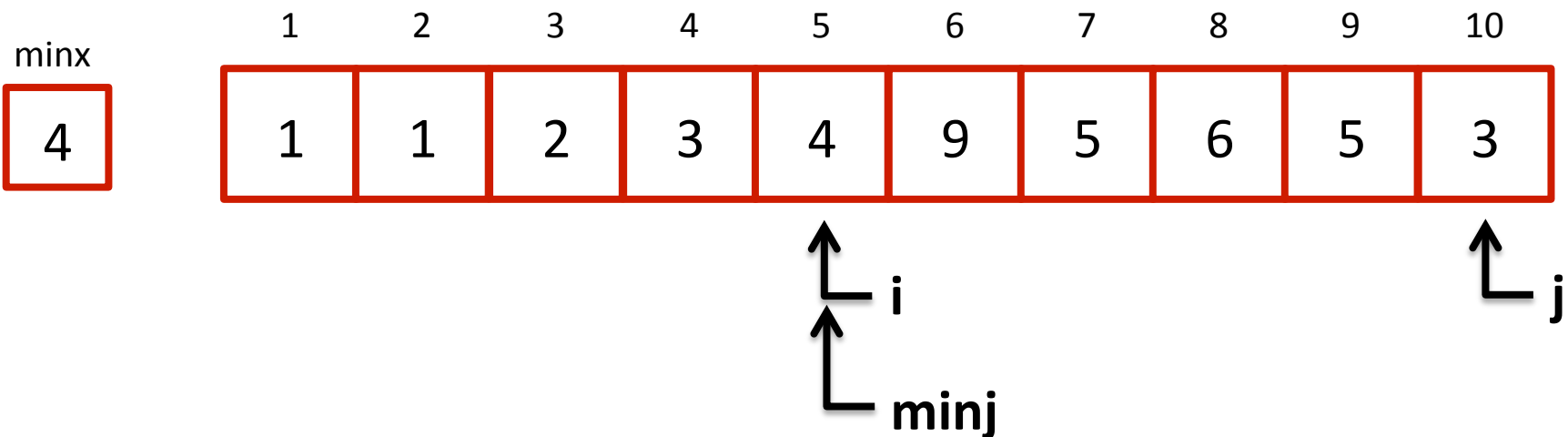
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

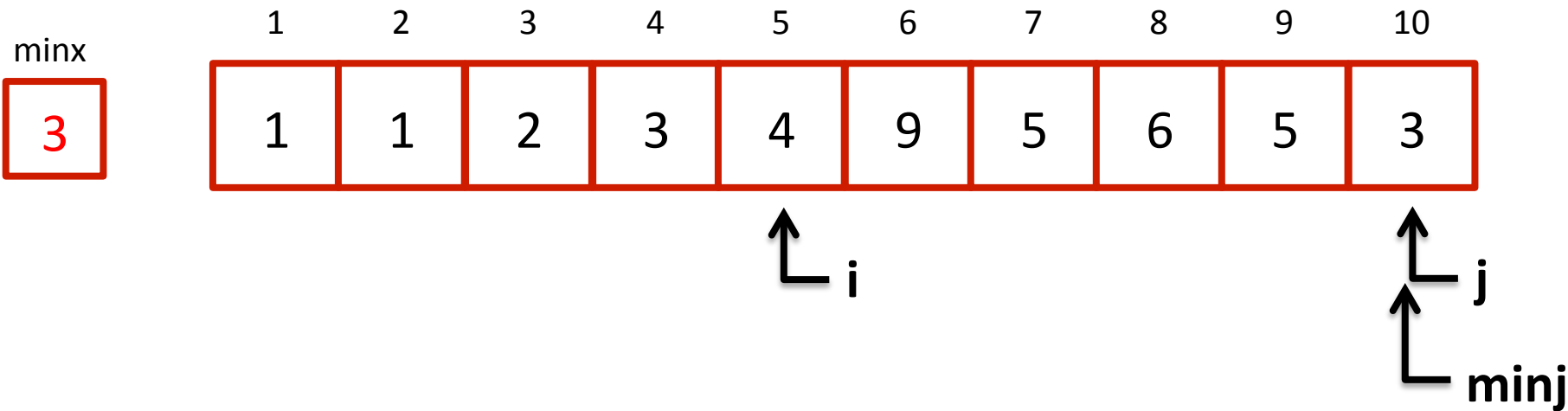
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

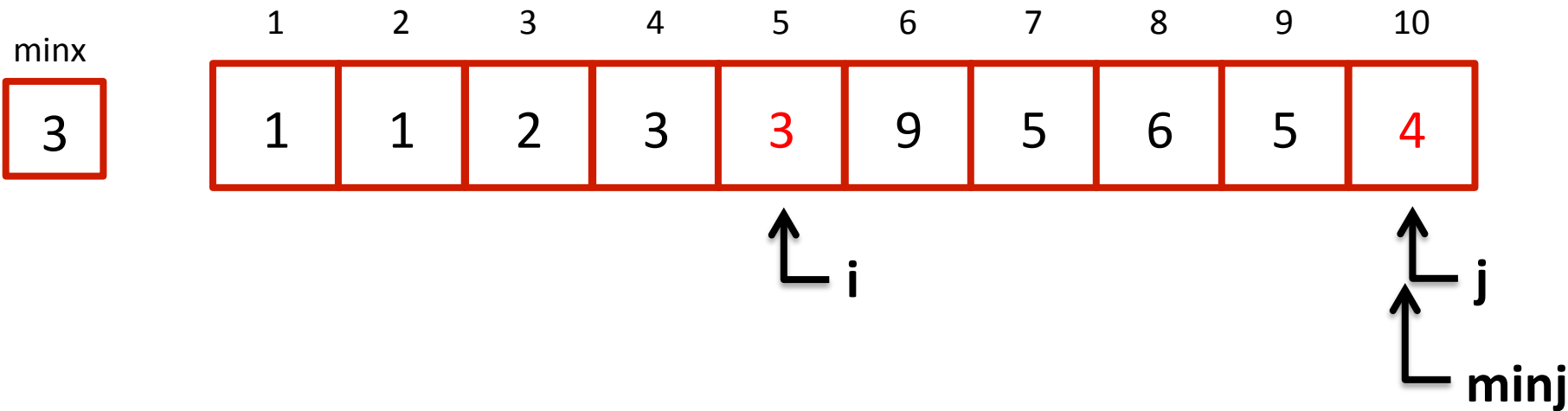
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

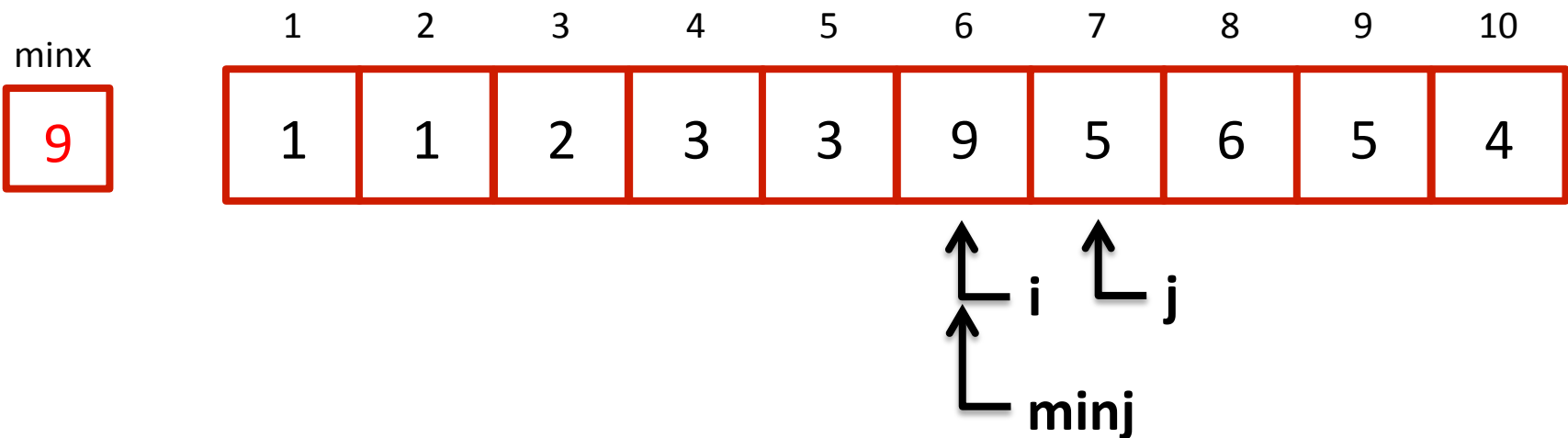
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

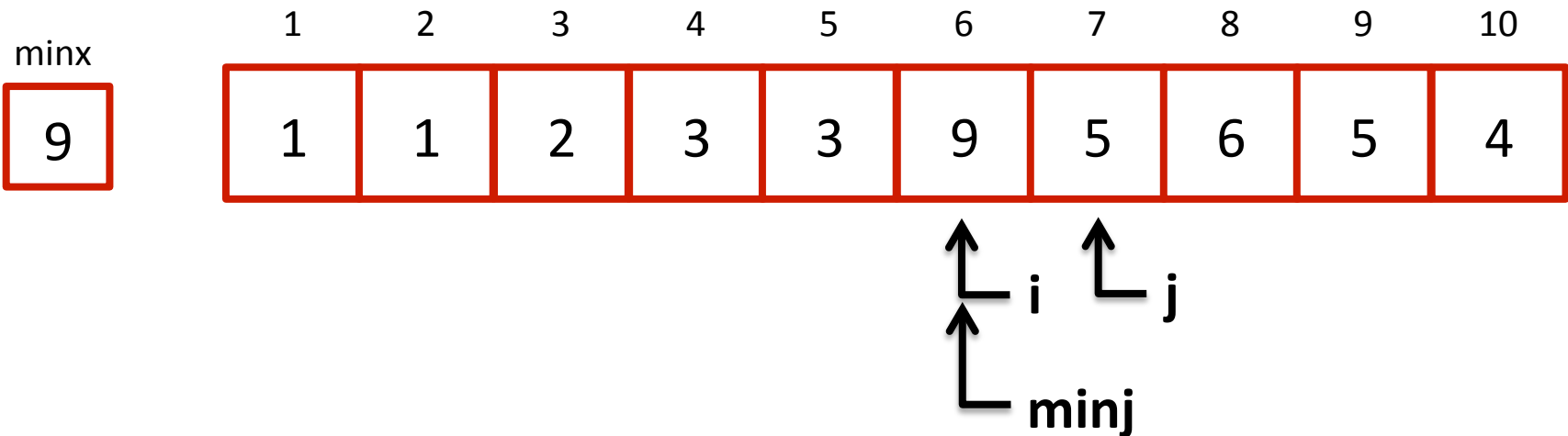
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

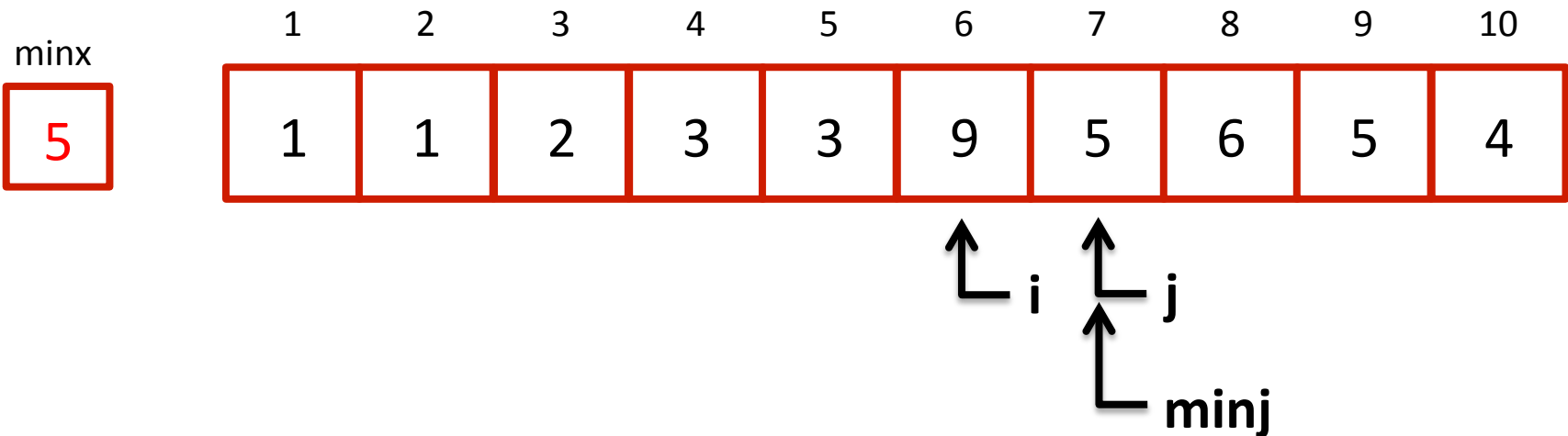
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

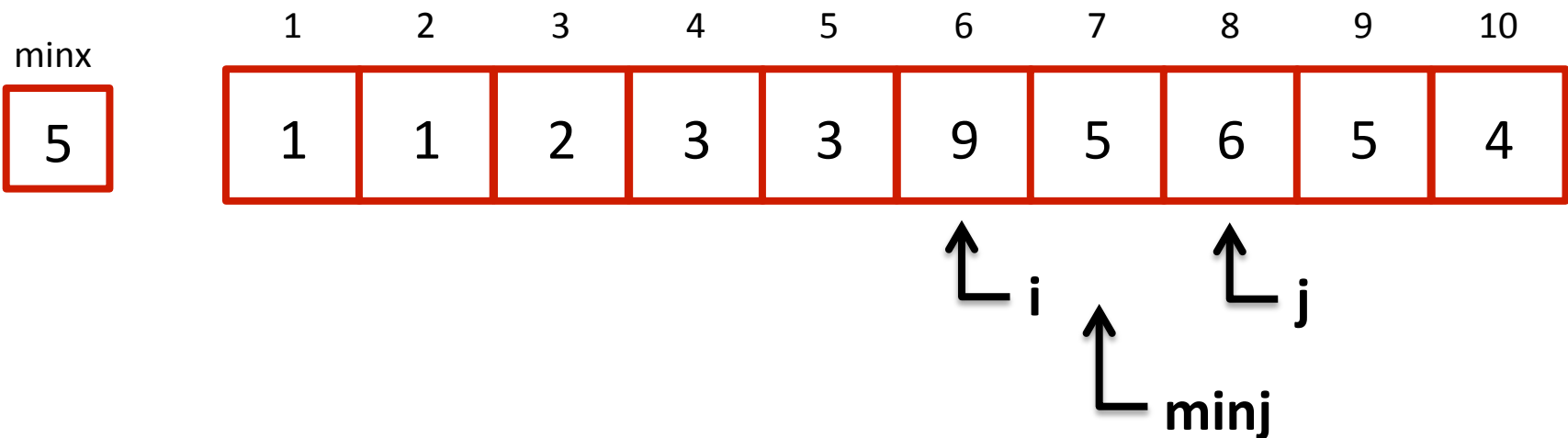
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

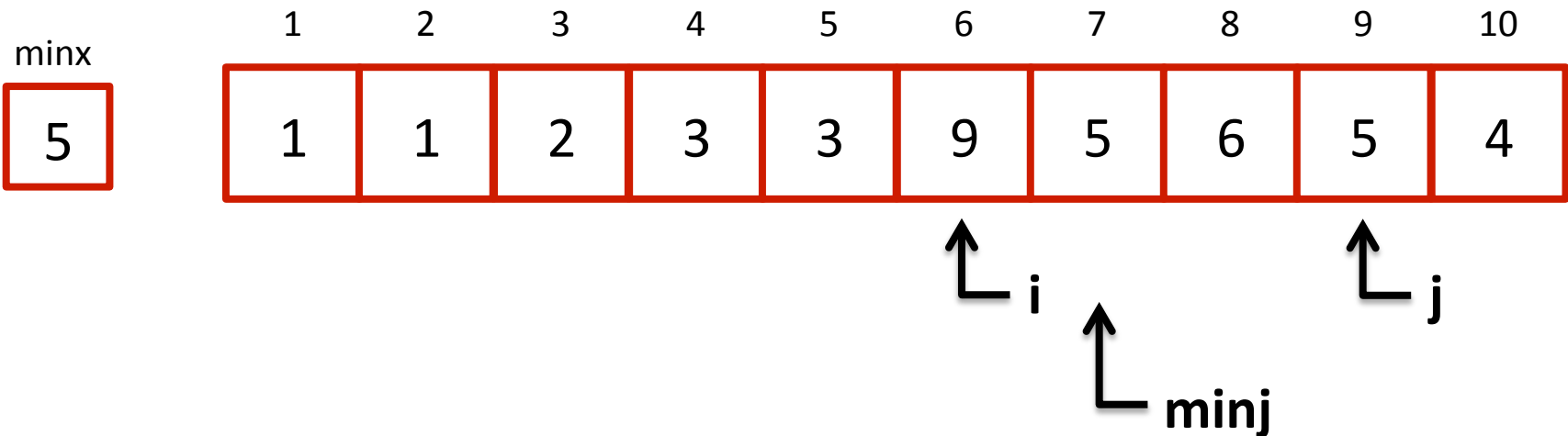
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

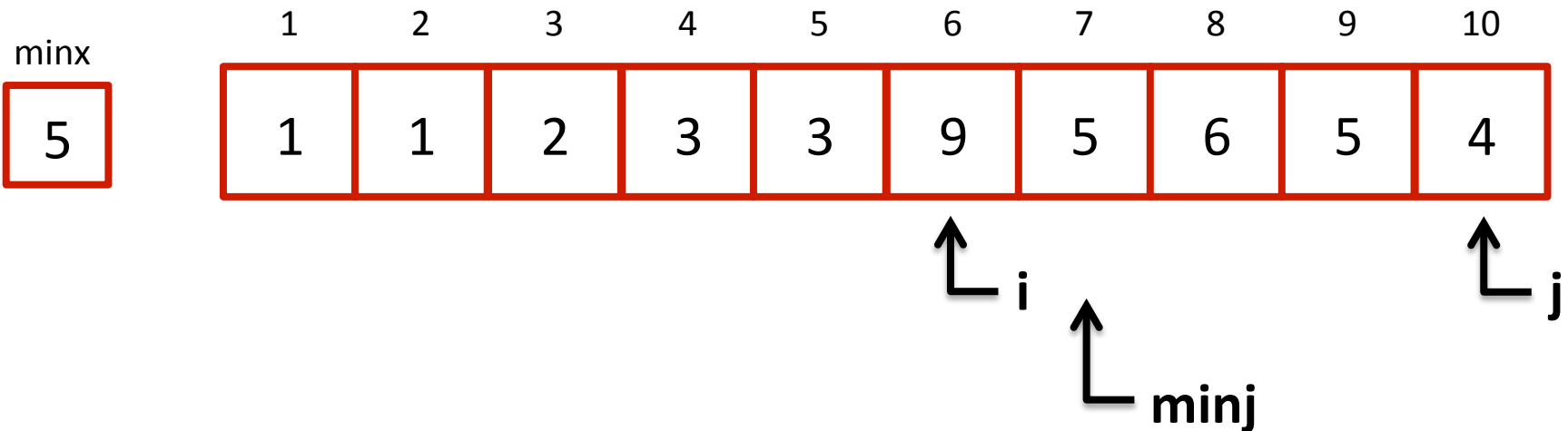
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

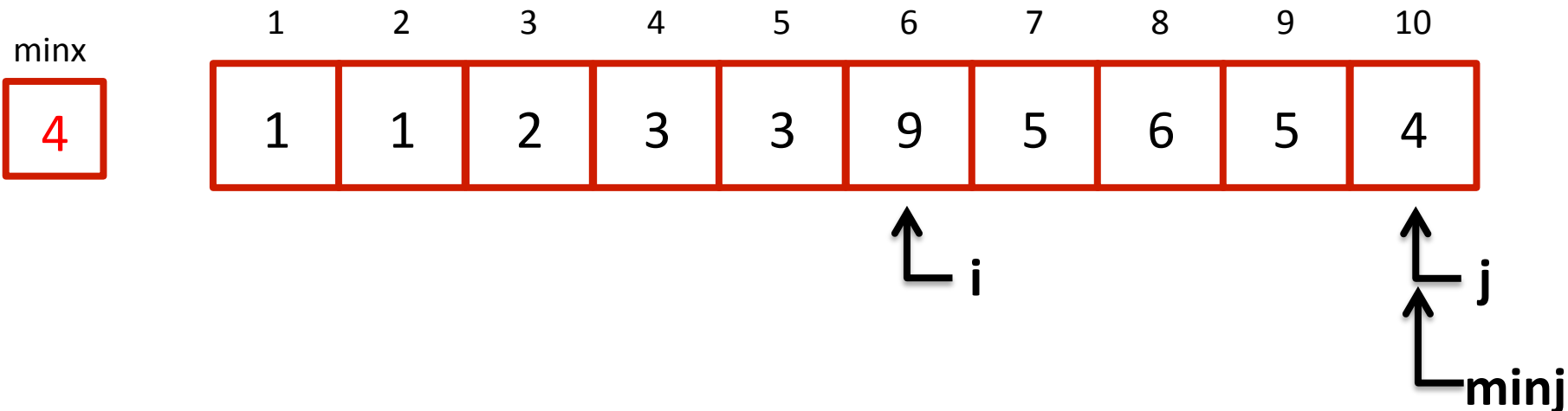
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

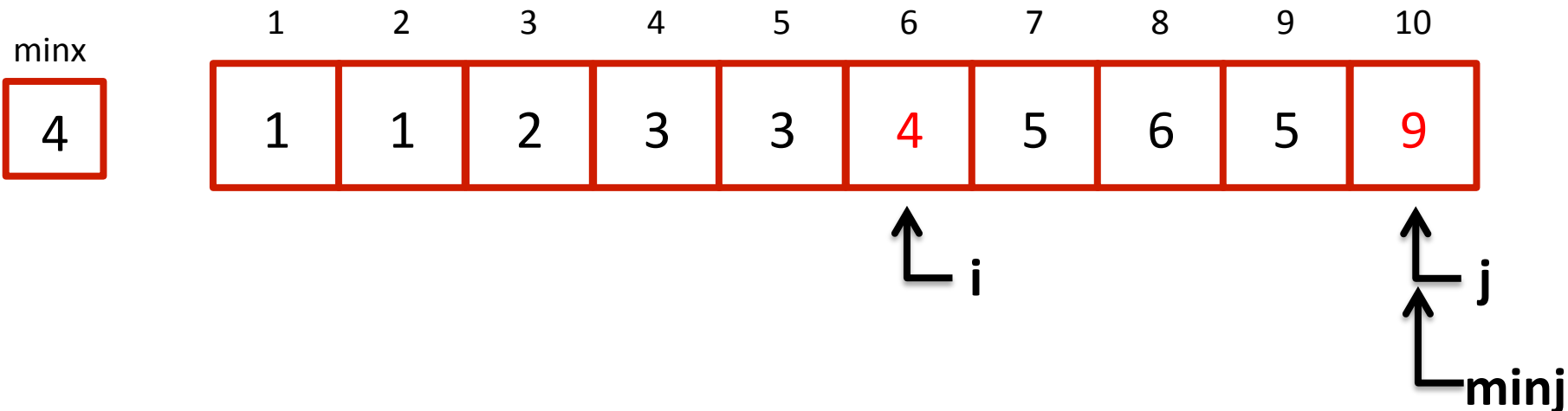
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

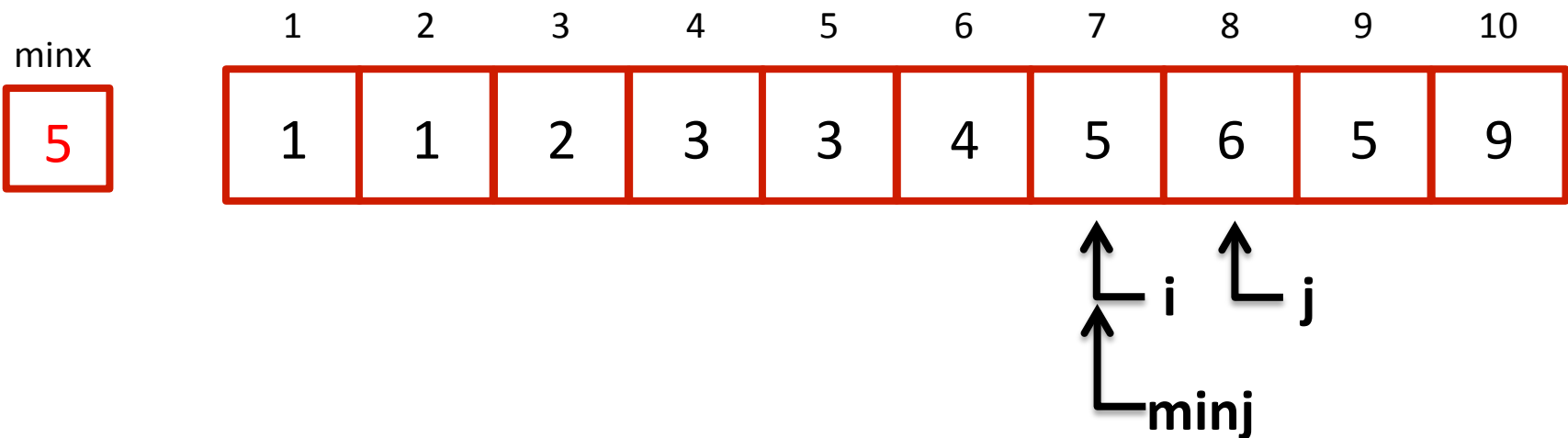
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

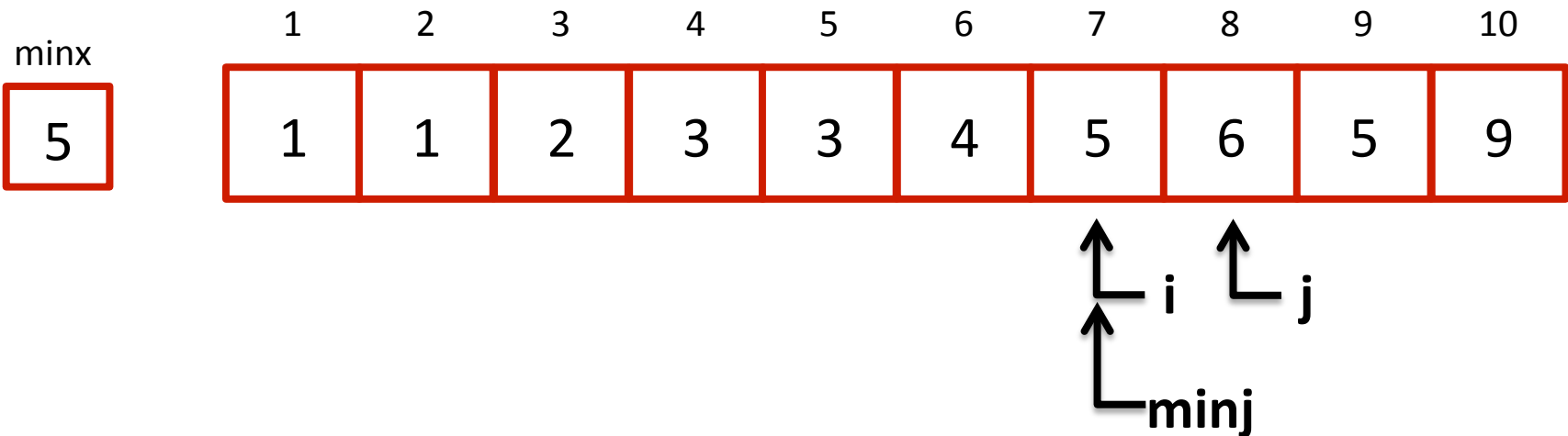
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

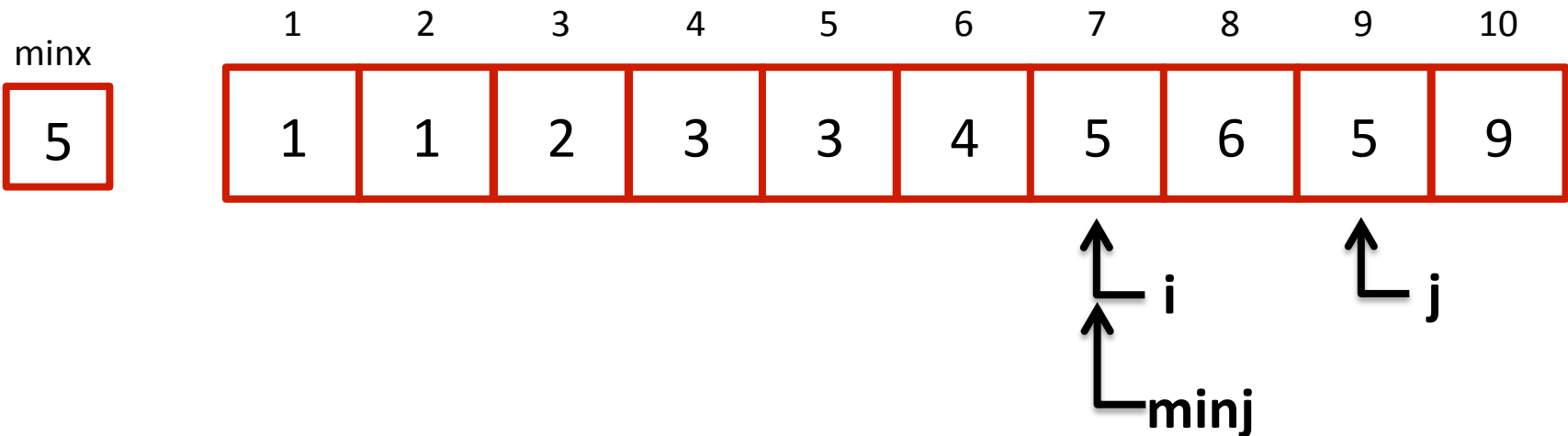
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

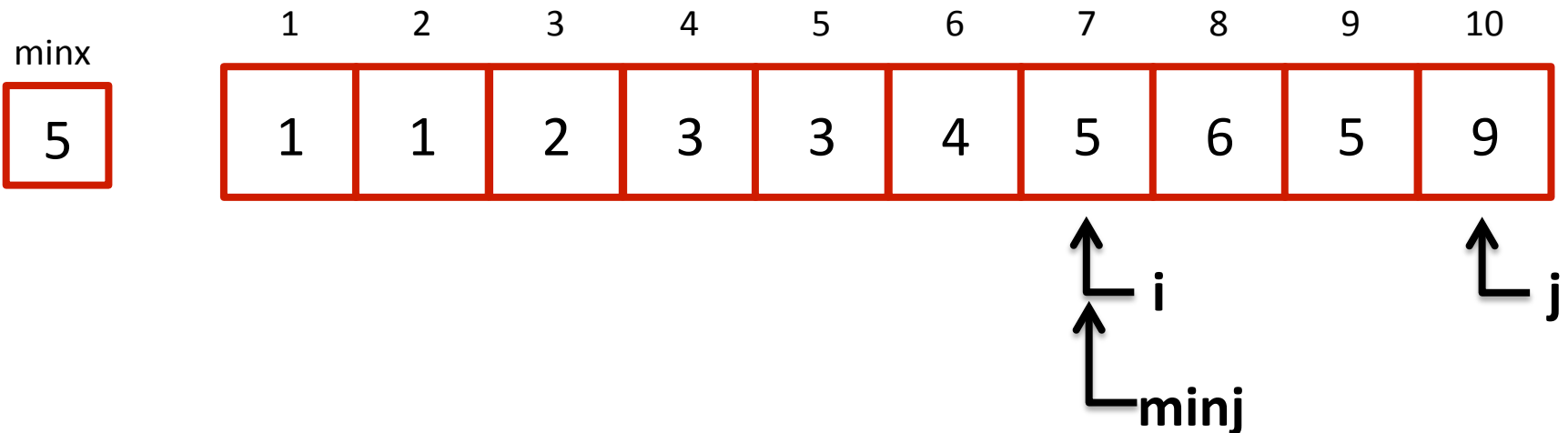
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

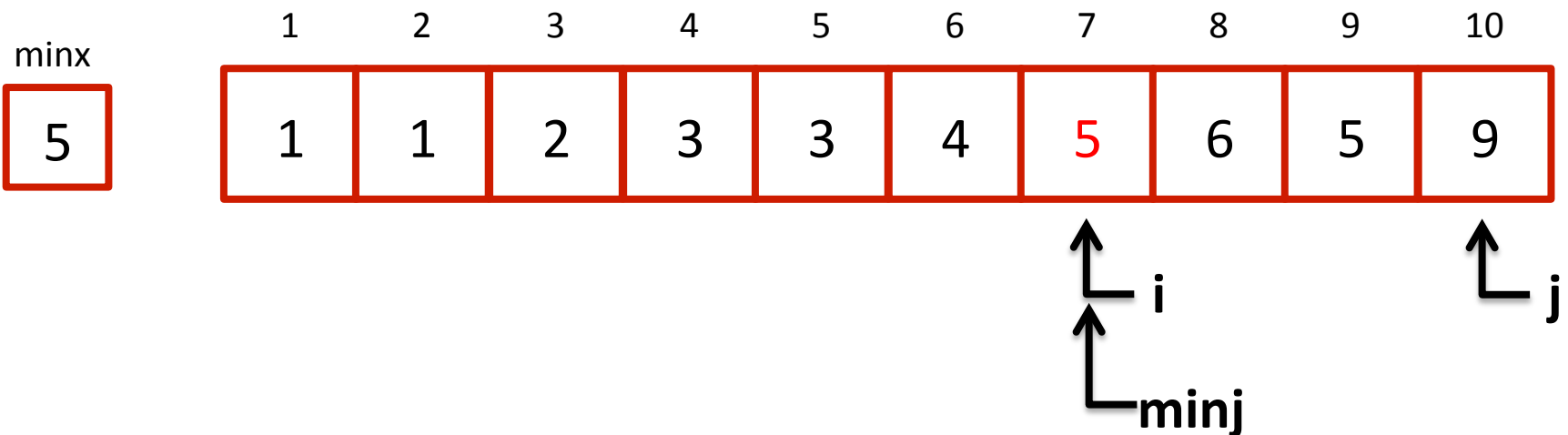
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

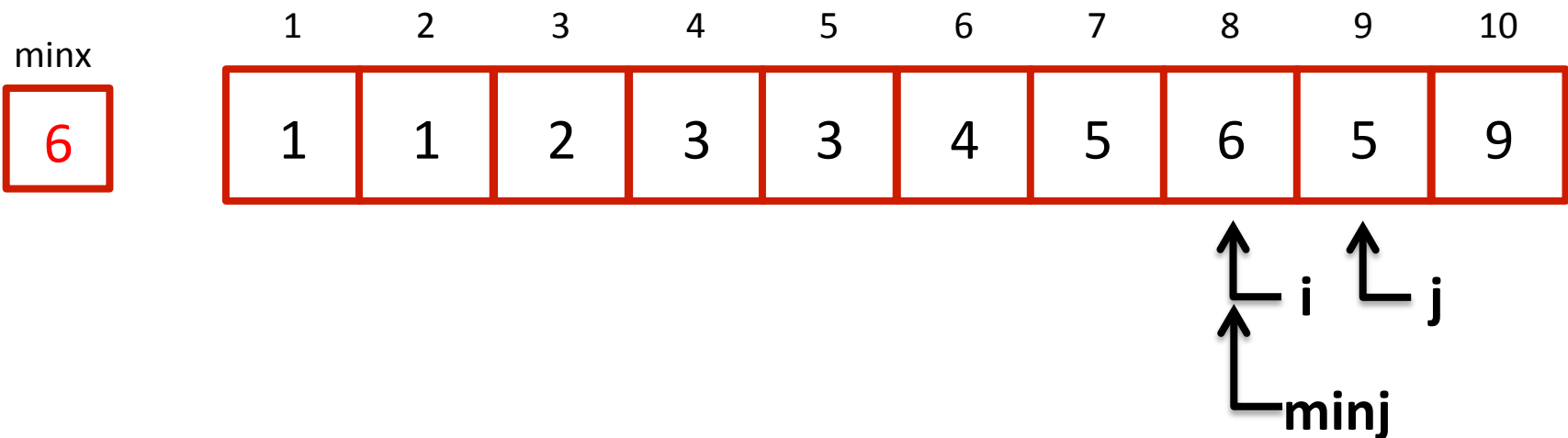
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

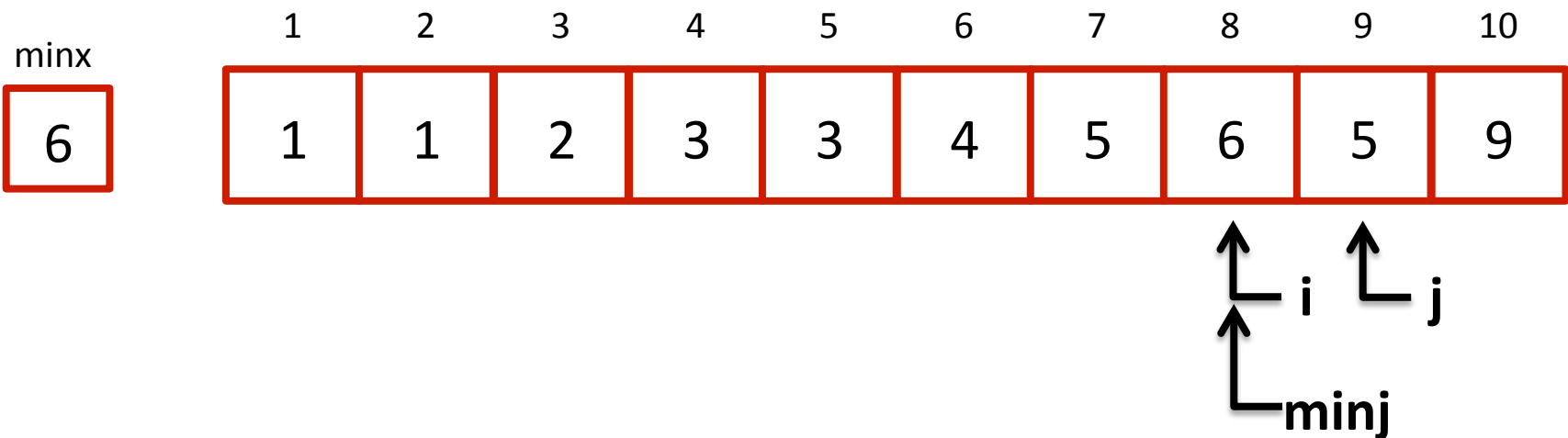
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

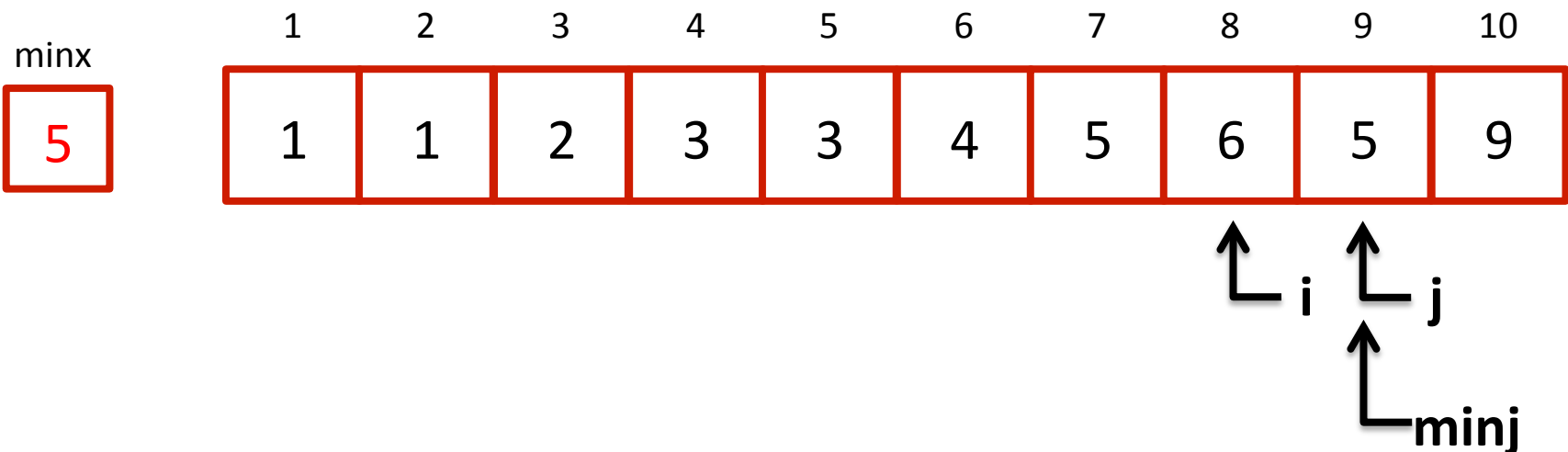
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

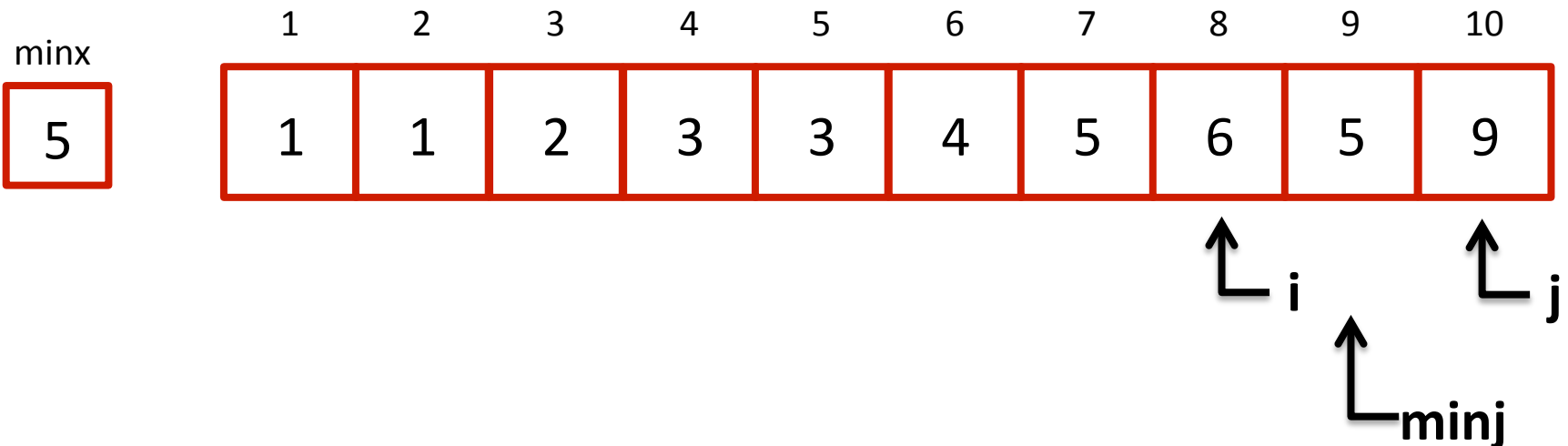
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

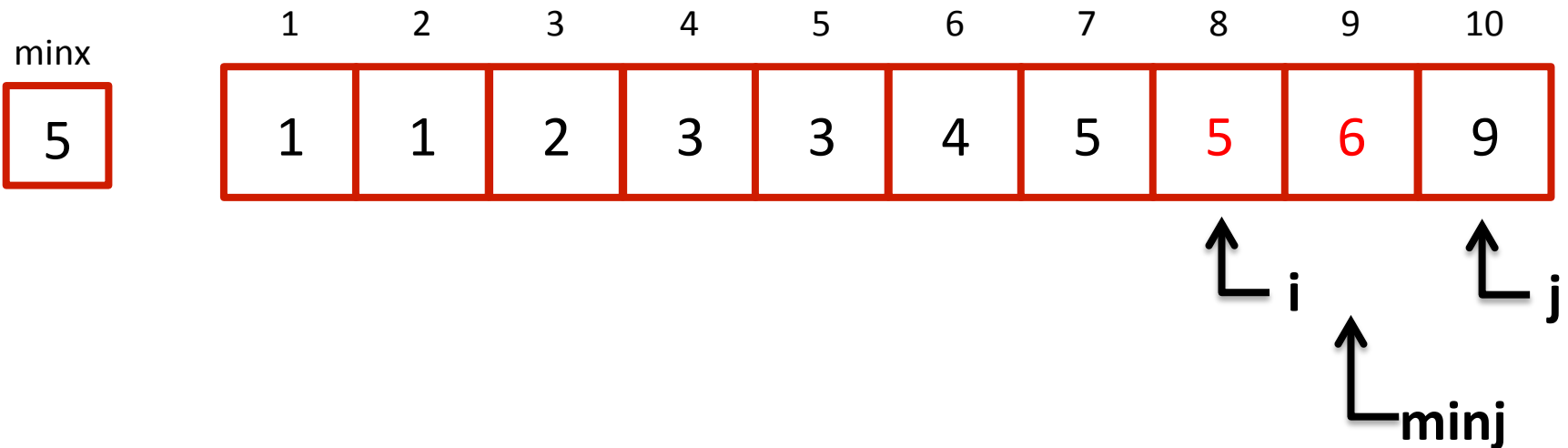
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

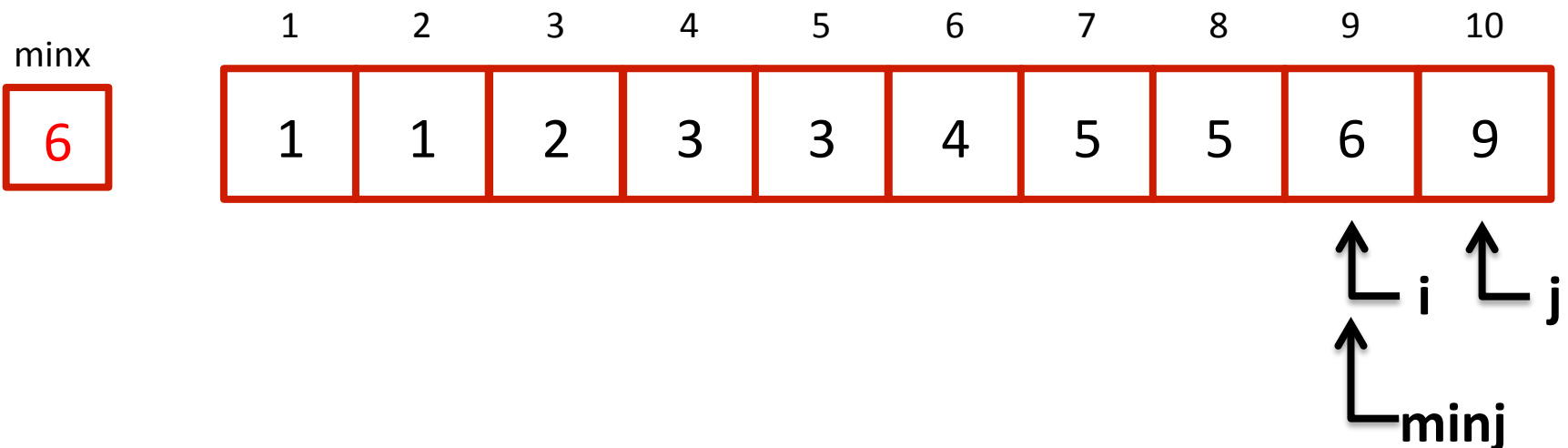
```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

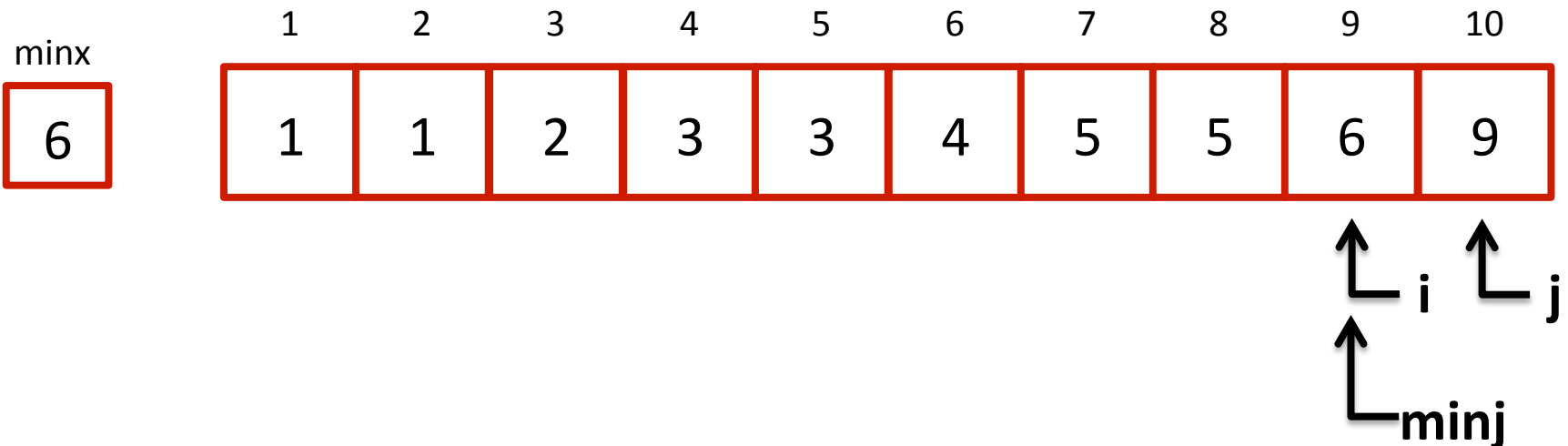
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

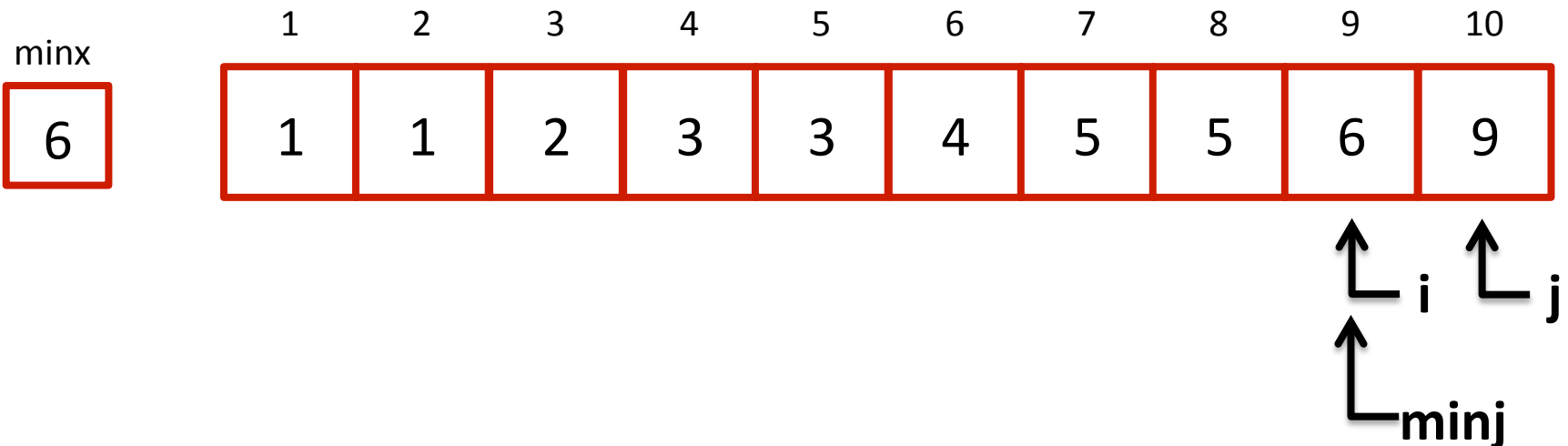
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])
```

```
  for i:=1 to n-1 do
```

```
    minj:=i;
```

```
    minx:=T[i];
```

```
    for j:=i+1 to n do
```

```
      if T[j]<minx then
```

```
        minj:=j;
```

```
        minx:=T[j]
```

```
      end if
```

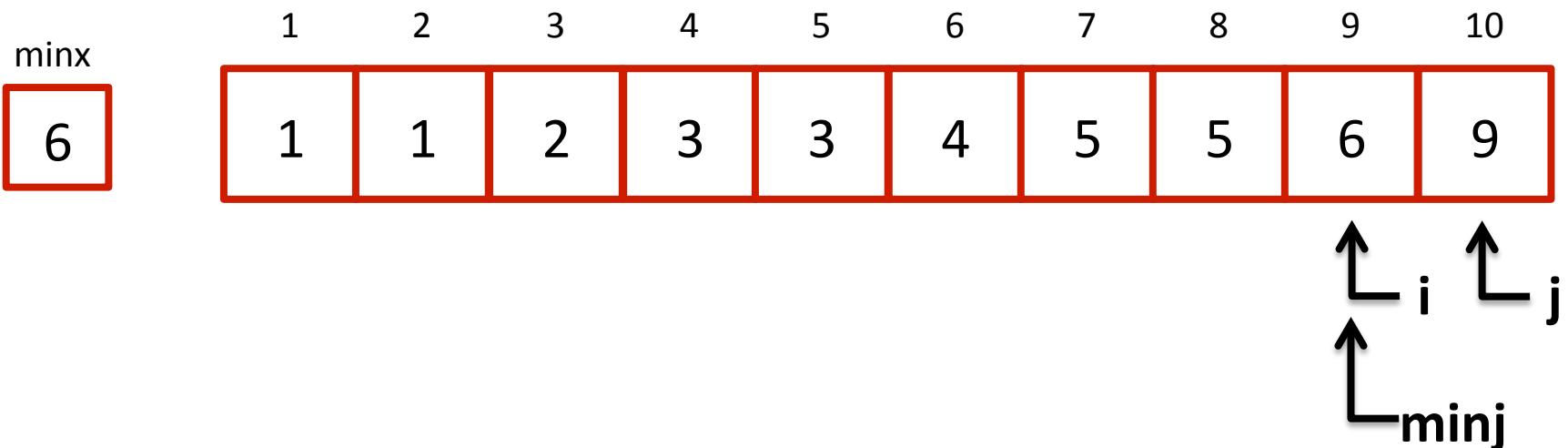
```
    end for;
```

```
    T[minj]:=T[i];
```

```
    T[i]:=minx
```

```
  end for
```

```
end procedure
```



Selection Sort

```
procedure Selection Sort (var T[1..n])  
  for i:=1 to n-1 do  
    minj:=i;  
    minx:=T[i];  
    for j:=i+1 to n do  
      if T[j]<minx then  
        minj:=j;  
        minx:=T[j]  
      end if  
    end for;  
    T[minj]:=T[i];  
    T[i]:=minx  
  end for  
end procedure
```

1	2	3	4	5	6	7	8	9	10
1	1	2	3	3	4	5	5	6	9