

Sort by Exchange

Sorting Arrays

- Assume that the items under consideration satisfy the following conditions for any values a , b , c :
 - Exactly one of the possibilities $a < b$, $a = b$, $a > b$ is true (law of trichotomy)
 - If $a < b$ and $b < c$, then $a < c$ (law of transitivity)

Sort in Non-Descending Order

A

7	9	8	2	5	2	6
----------	----------	----------	----------	----------	----------	----------

2	2	5	6	7	8	9
----------	----------	----------	----------	----------	----------	----------

Approach

- Find the smallest element in the array
- Move the element to the beginning of A

Approach

The following construct will find the minimum value in the array:

```
min = A[0]
j = 1
While j < N
    If A[j] < min
        min = A[j]
    EndIf
    j = j + 1
EndWhile
```

We can place min in A[0] with:

```
A[0] = min
```

Approach

Performing these two steps on the array we get:

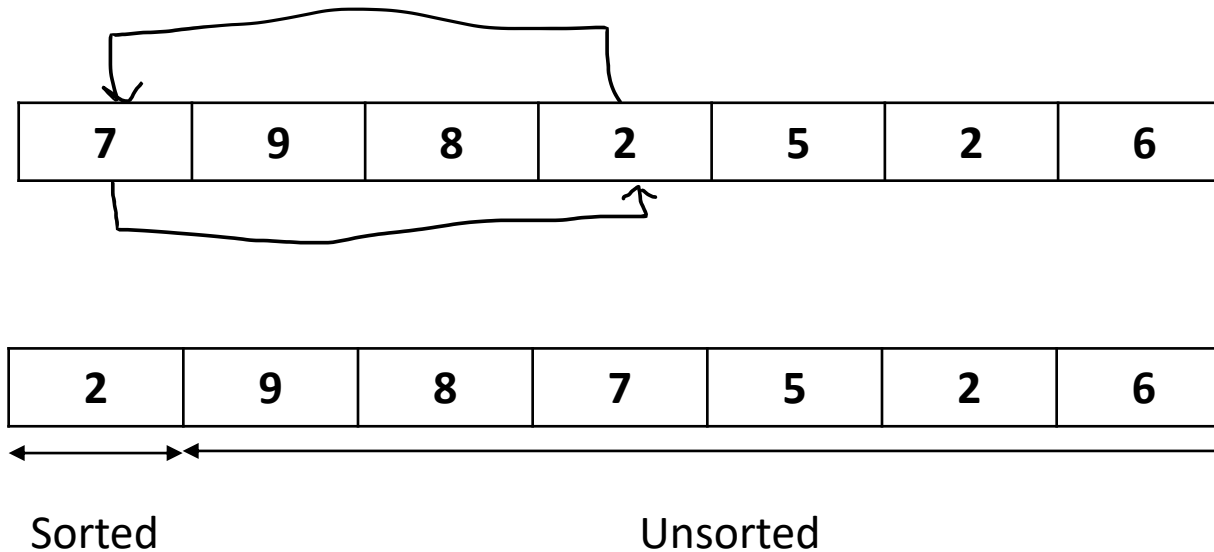
7	9	8	2	5	2	6
---	---	---	---	---	---	---

2	9	8	2	5	2	6
---	---	---	---	---	---	---

This leaves us with our old problem – we've by moving one element we overwritten one element. We need to remember the one we replaced. We can store it in a temporary variable.

Approach

We can swap elements – splitting the array into a sorted and unsorted part.
To do this we need to not only find the minimum but also remember where it is.



```
min = A[0]
j = 1
p = 0
While j < N
    If A[j] < min
        min = A[j]
        p = j
    EndIf
    j = j + 1
EndWhile
A[p] = A[0]
A[0] = min
```


We can do this for the unsorted part of the Array [A[1] and up]

```
min = A[1]
j = 2
p = 1
While j < N
    If A[j] < min
        min = A[j]
        p = j
    EndIf
    j = j + 1
EndWhile
A[p] = A[1]
A[1] = min
```

Can we make that into a loop?

As a loop

Program SortExchange

Read A

i=0

While i<N-1

min = A[i]

j = i+1

p = i

While j<N

If A[j]<min

min = A[j]

p = j

EndIf

j = j+1

EndWhile

A[p] = A[i]

A[i] = min

i = i+1

EndWhile

End