

Sorting & 1D Arrays

ICS 3M0 - Java Programming
Sort ... Modified Bubble Sort

Sorting & 1D Arrays

Sorting & 1D Arrays

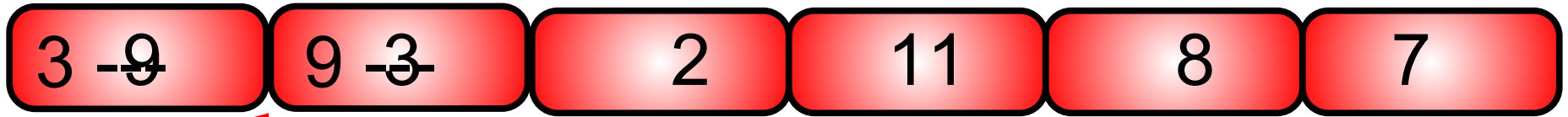
Sorting:

- placing items in a list (array) in order

 Ascending Order: A-Z, 0-9

 Descending Order: Z-A, 9-0

Sorting & 1D Arrays



You can't just exchange the values ... this would be

temp

Destructive

Write We will use a

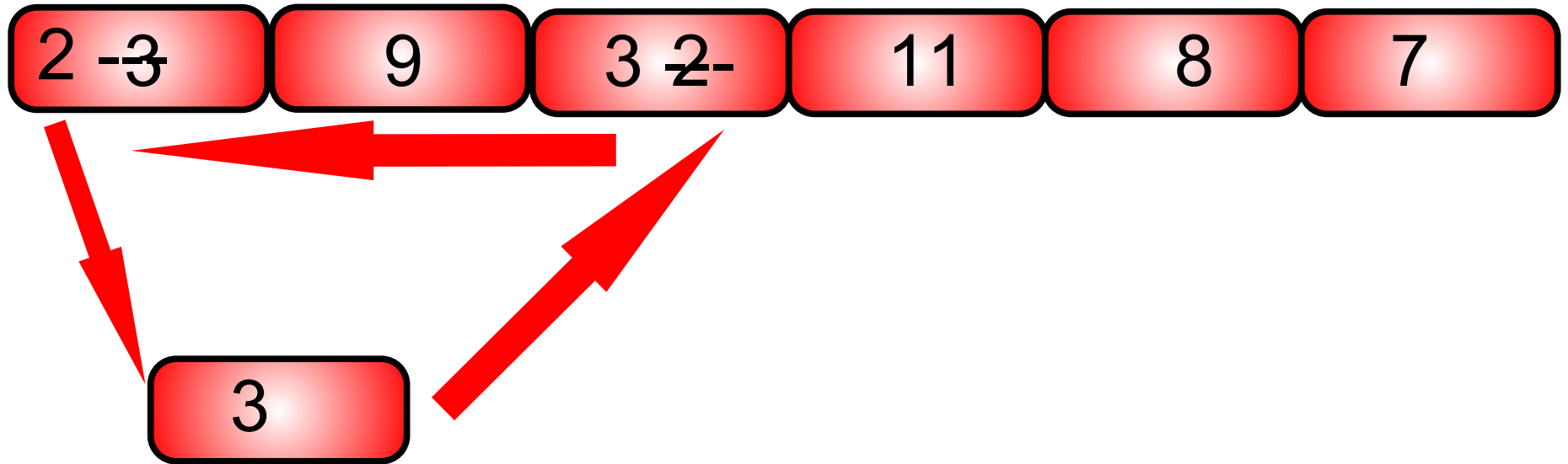
temporary memory location. We can re-use it again

ie $A = 9$, $B = 3$,
 $A = B$, - would other swap ...

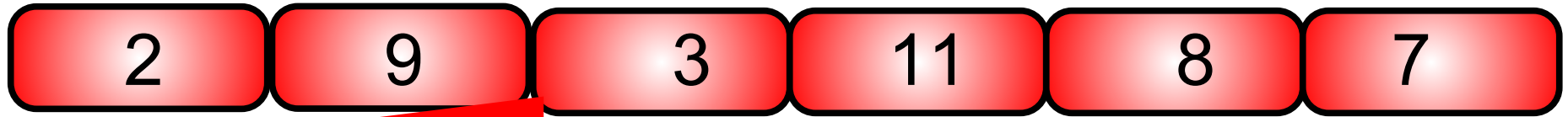
to 'swap' the

copy 3 to A,
data
and erase 9!

Sorting & 1D Arrays



Sorting & 1D Arrays



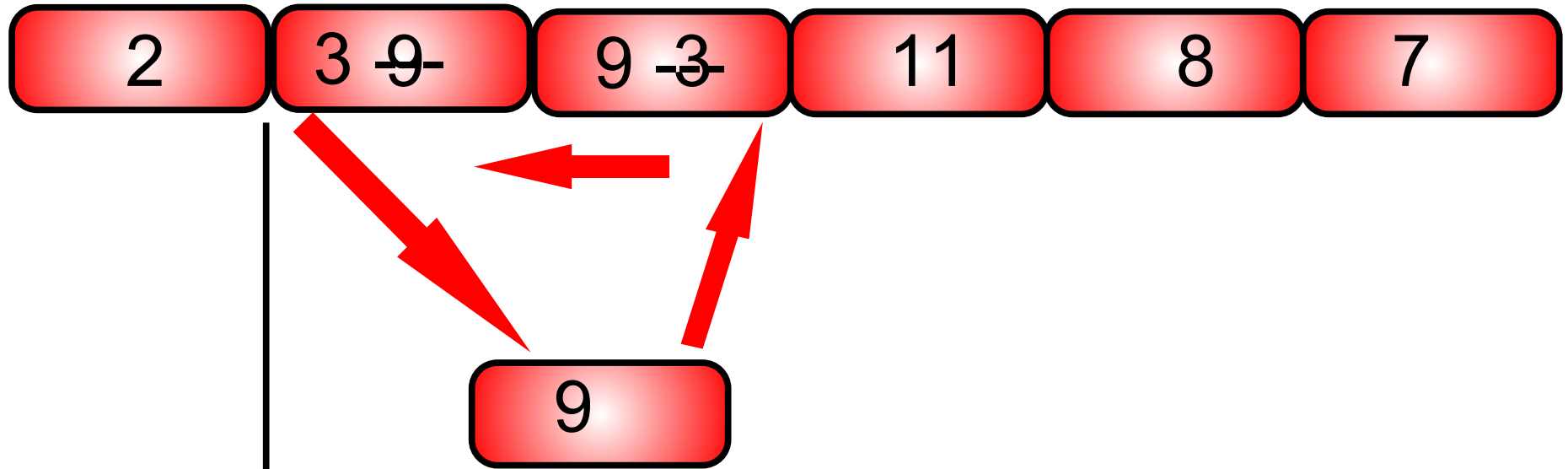
2 is smaller than 11,
therefore, no swap
necessary!

2 is smaller than 8, thus
no swap!

2 is smaller than 7,
thus no swap!

The 2 has
'bubbled' to the
top. It is now
guaranteed to be
the smallest value!

Sorting & 1D Arrays

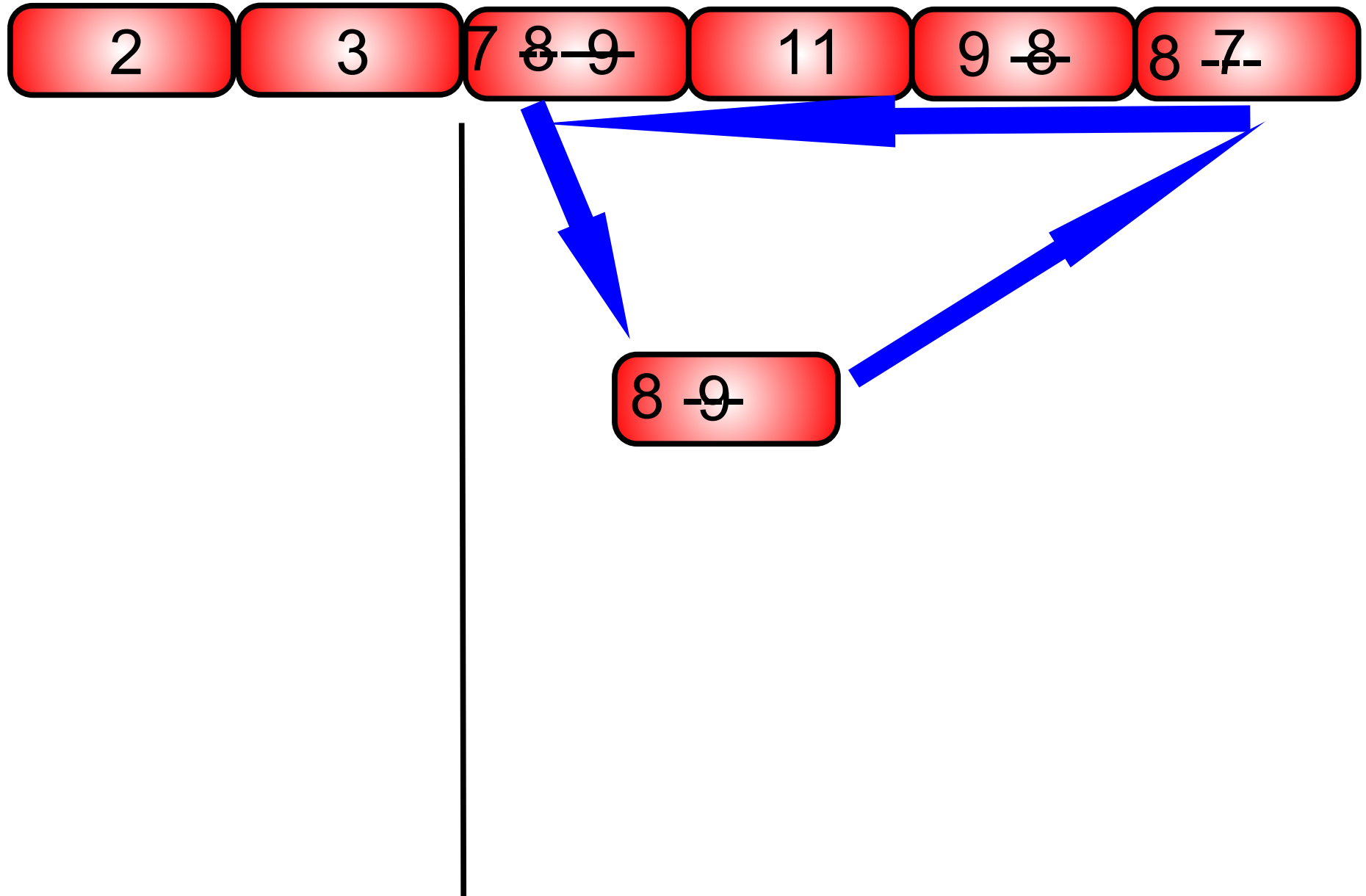


3 is less than 11

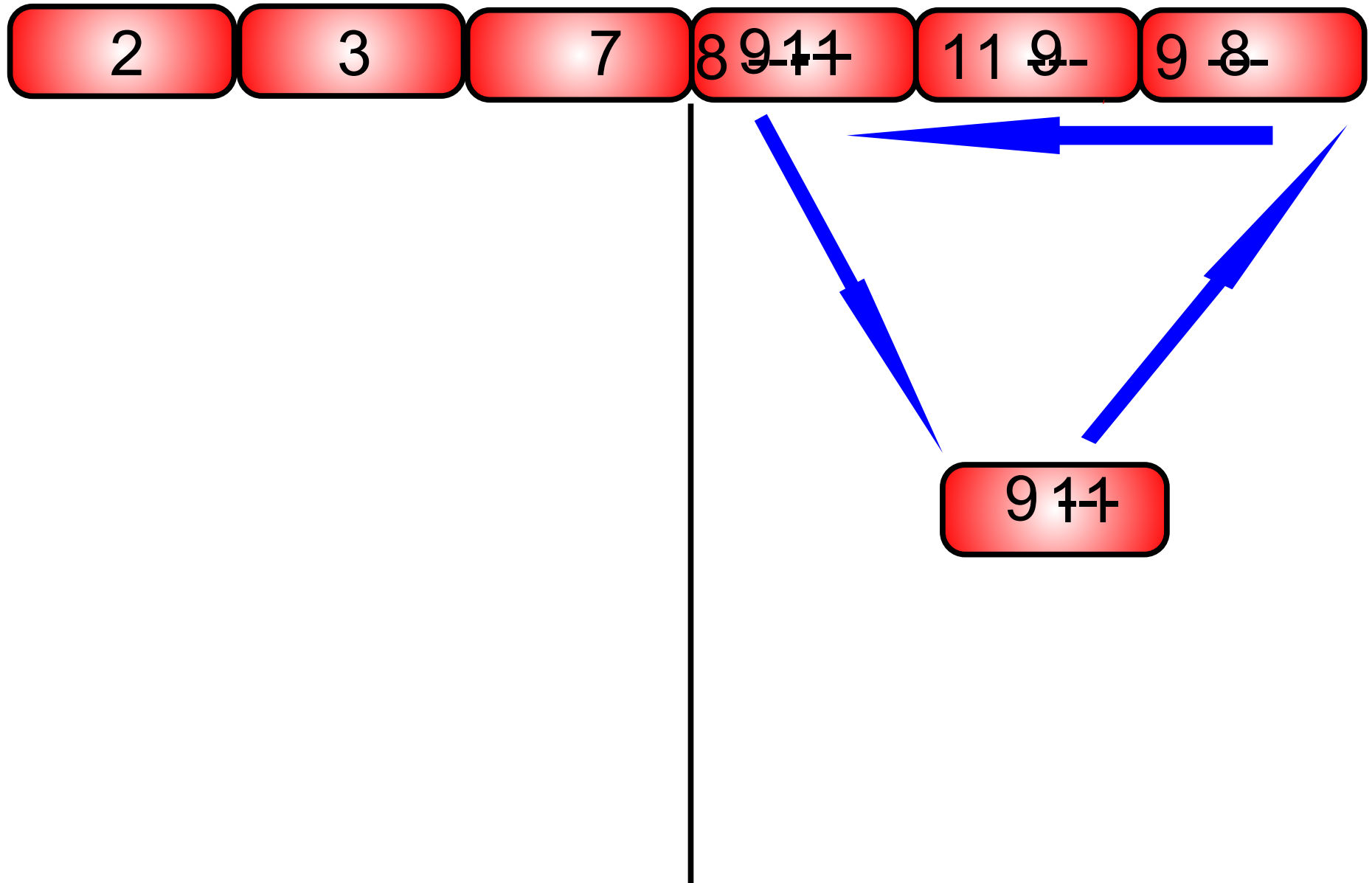
3 is less than 8

3 is less than 7

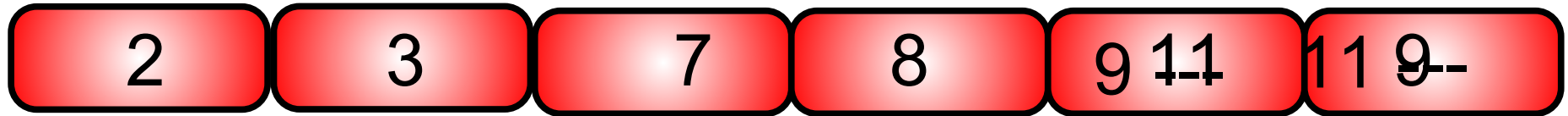
Sorting & 1D Arrays



Sorting & 1D Arrays



Sorting & 1D Arrays



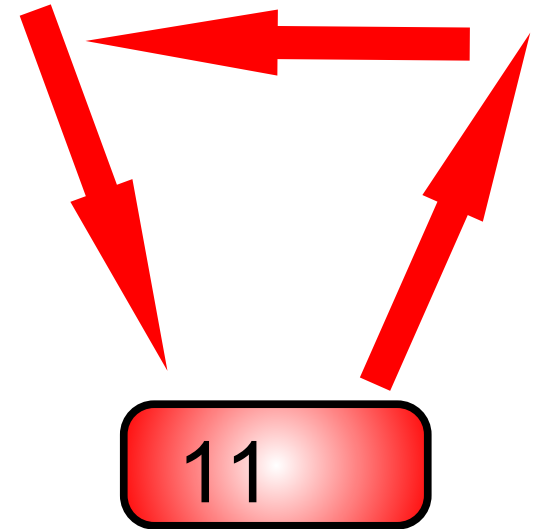
- the SWAP is made
IF the first value is greater than
the second

- the first swap goes from
elements 0 to max-1 (one less
than the maximum elements)

ie: $x = 0$ to max-1

- the last swap goes from 1
(always starts 1 above the first
swap element) to max

ie: $y = x+1$ to max



Notice: After
the last swap,
everything is in
order!

Sorting & 1D Arrays

```
for (int x=0; x<max-1; x++)
{
    for (int y=x+1; y<max; y++)
    {
        // 'if' determines the field to sort by (item) and the order
        // (ascending/ // descending indicated by < or >)
        if (item[x] < item[y])
        {
            // swap
            temp = item[x];    // notice the last/first pattern in
            item[x] = item[y]; // the swap
            item[y] = temp;
        }
    }
}
```

Sorting & 1D Arrays

What happens when we have related lists?

Name

Vaswani xxxxx Ankit
Ankitxxxxxxx Vaswani

Pay

7.99
24.99

When we sort the name ...

Vaswani is very happy ... but Ankit is not

Solution?

Swap both lists!

Sorting & 1D Arrays

```
for (int x = 0; x < max-1; x++)
{
    for (int y = x+1; y < max; y++)
    {
        if (name[x].compareTo(name[y]) > 0)
        {
            tempString = name[x];
            name[x] = name[y];
            name[y] = tempString;
            tempDouble = pay[x];
            pay[x] = pay[y];
            pay[y] = tempDouble;
        }
    }
}
```

Note: compareTo returns <0, 0, or >0

<0 means that the first string is less than the second

= means that the two strings compare

>0 means that the first string is greater than the second