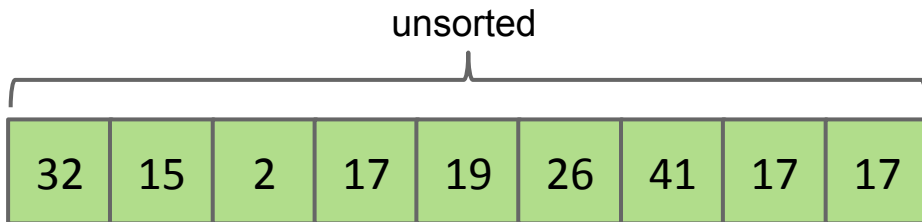


Top-Down Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half.
- Merge the two sorted halves to form the final result.

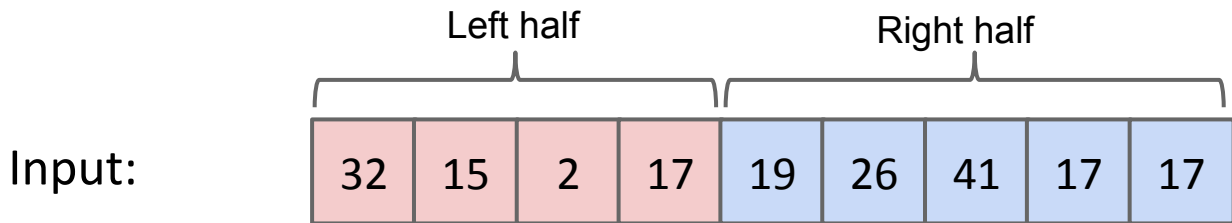
Input:



Top-Down Merge Sort

Top-Down merge sorting N items:

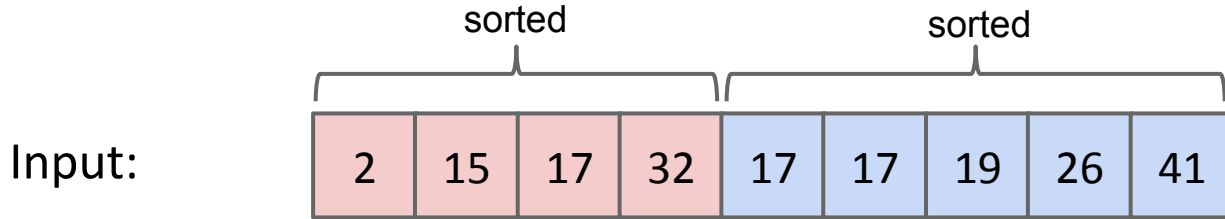
- **Split items into 2 roughly even pieces.**
- Mergesort each half.
- Merge the two sorted halves to form the final result.



Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- **Mergesort each half (steps not shown, this is a recursive algorithm!)**
- Merge the two sorted halves to form the final result.



Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- **Merge the two sorted halves to form the final result.**
 - Compare $\text{input}[i] < \text{input}[j]$.
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
	i			j				

Aux:

2	0	0	0	0	0	0	0	0
	p							

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
	i			j				

Aux:

2	0	0	0	0	0	0	0	0
	p							

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
		i		j				

Aux:

2	15	0	0	0	0	0	0	0
		p						

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
		i		j				

Aux:

2	15	0	0	0	0	0	0	0
		p						

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i j

Aux:

2	15	17	0	0	0	0	0	0
---	----	----	---	---	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i j

Aux:

2	15	17	0	0	0	0	0	0
---	----	----	---	---	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	0	0	0	0	0
---	----	----	----	---	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
			i		j			

Aux:

2	15	17	17	0	0	0	0	0
				p				

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	0	0	0	0
---	----	----	----	----	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	0	0	0	0
---	----	----	----	----	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	0	0	0	0
---	----	----	----	----	---	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	19	0	0	0
---	----	----	----	----	----	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	19	0	0	0
---	----	----	----	----	----	---	---	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
			i					j

Aux:

2	15	17	17	17	19	26	0	0
							p	

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

Input:

2	15	17	32	17	17	19	26	41
			i					j

Aux:

2	15	17	17	17	19	26	0	0
							p	

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - **Compare $\text{input}[i] < \text{input}[j]$ (if necessary).**
 - Copy smaller item and increment p and i or j.

No comparison is made
this time, since the left
side has run out of items!

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	19	26	32	0
---	----	----	----	----	----	----	----	---

p

Merge Sort

Top-Down merge sorting N items:

- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
 - Compare $\text{input}[i] < \text{input}[j]$ (if necessary).
 - **Copy smaller item and increment p and i or j.**

Input:

2	15	17	32	17	17	19	26	41
---	----	----	----	----	----	----	----	----

i

j

Aux:

2	15	17	17	17	19	26	32	41
---	----	----	----	----	----	----	----	----

p