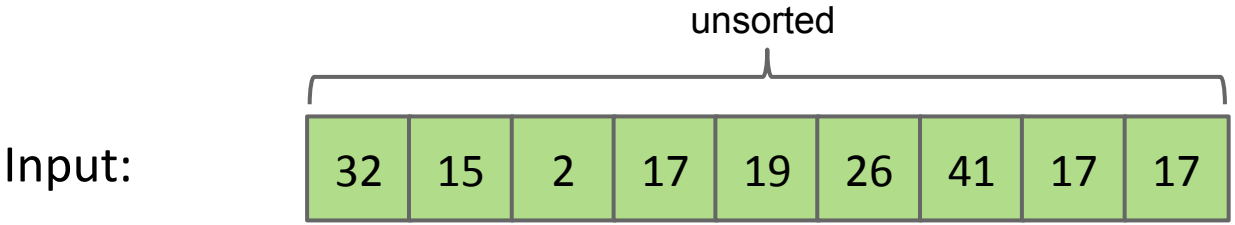
Top-Down Merge Sort

Top-Down merge sorting N items:

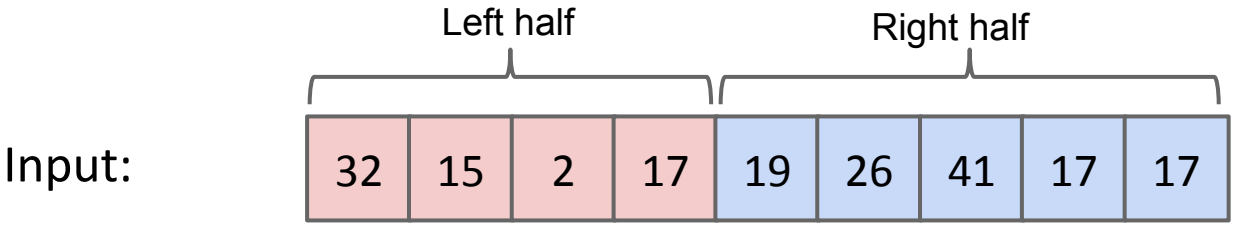
- Split items into 2 roughly even pieces.
- Mergesort each half.
- Merge the two sorted halves to form the final result.



Top-Down Merge Sort

Top-Down merge sorting N items:

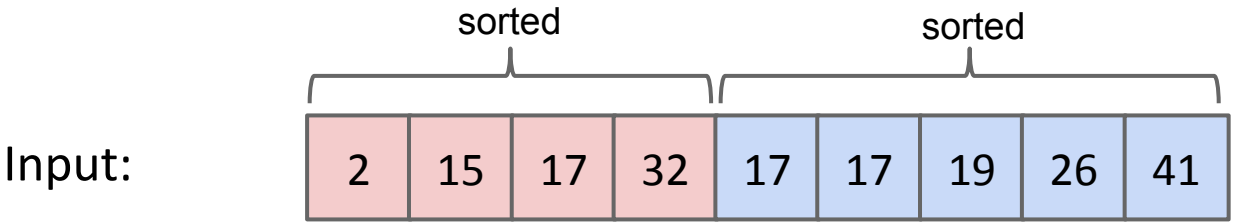
- **Split items into 2 roughly even pieces.**
- Mergesort each half.
- Merge the two sorted halves to form the final result.



Merge Sort

Top-Down merge sorting N items:

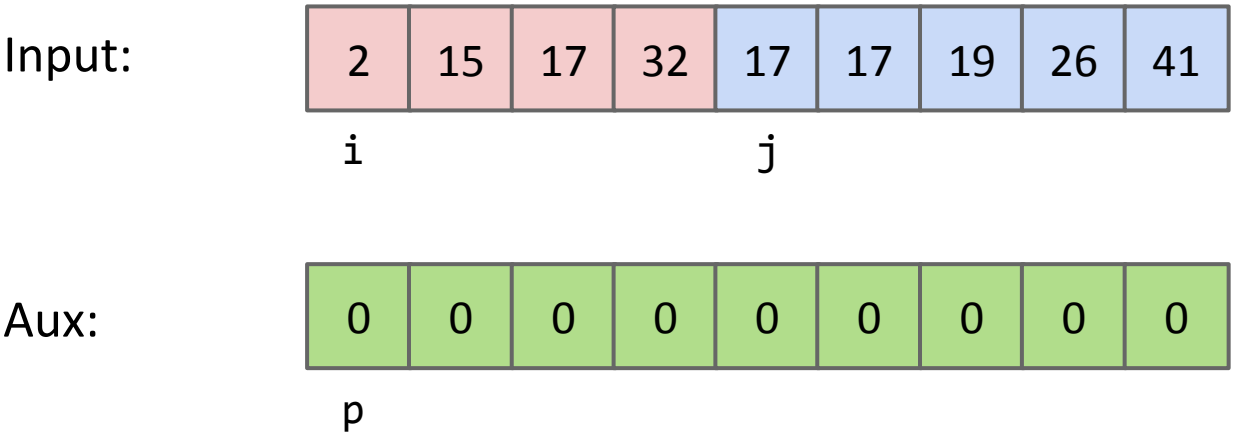
- Split items into 2 roughly even pieces.
- **Mergesort each half (steps not shown, this is a recursive algorithm!)**
- Merge the two sorted halves to form the final result.



Merge Sort

Top-Down merge sorting N items:

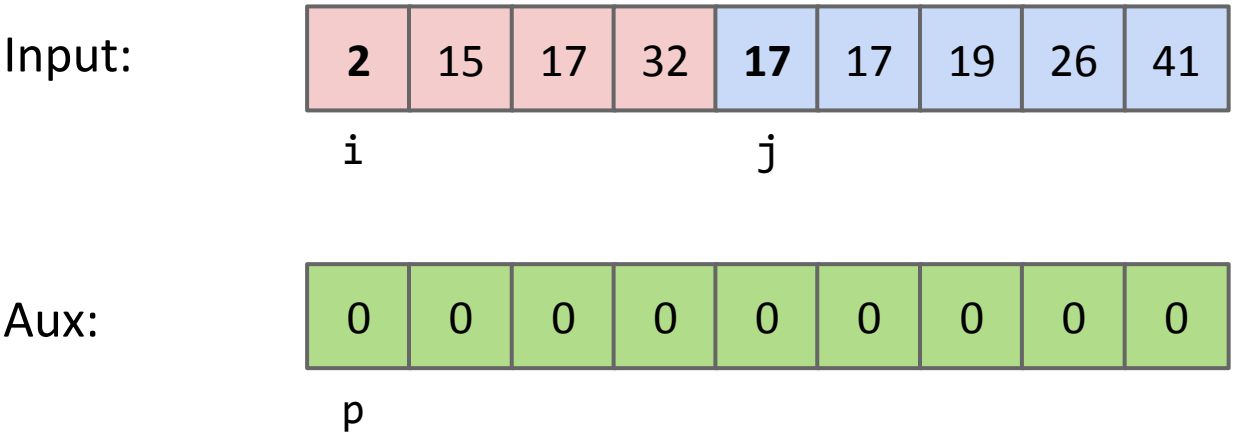
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- **Merge the two sorted halves to form the final result.**
  - Compare input[i] < input[j].
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

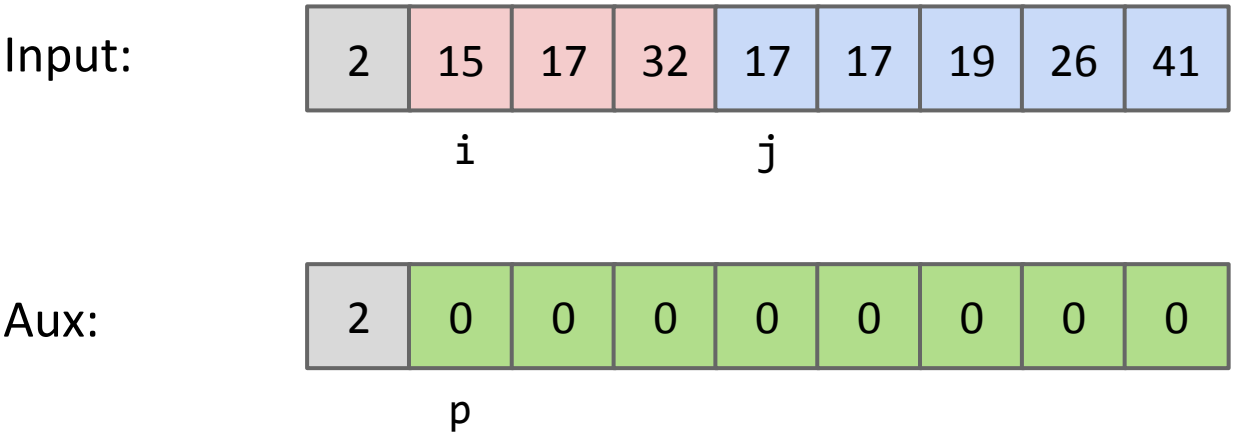
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

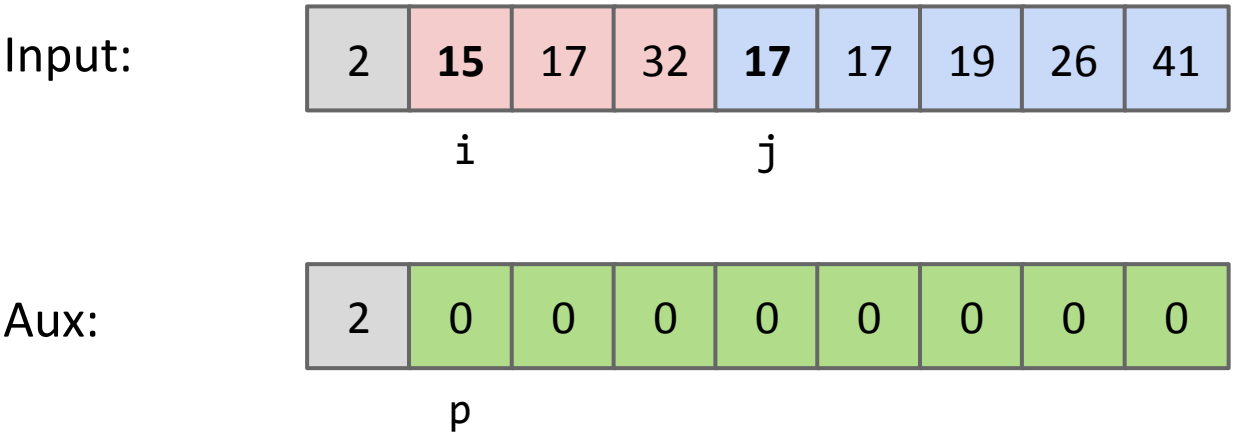
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

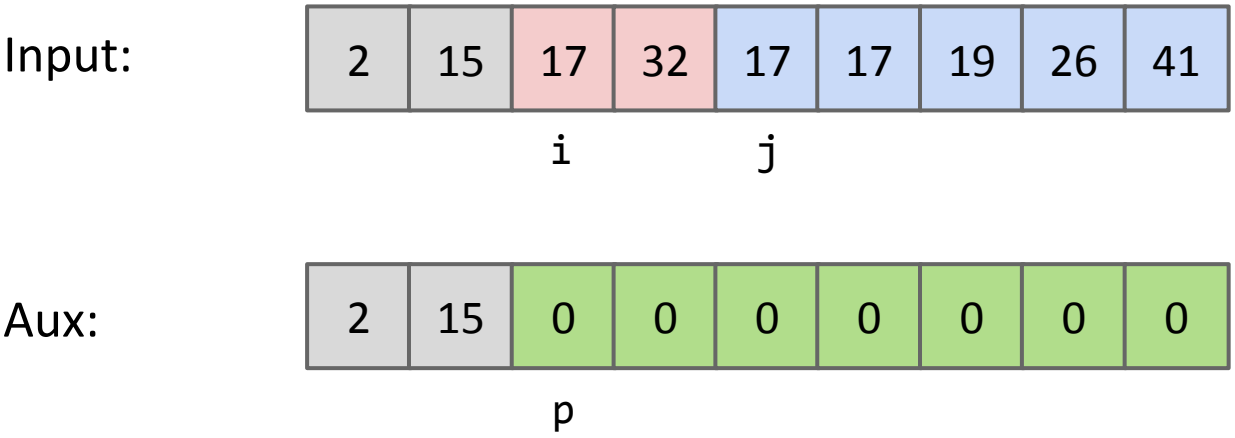
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

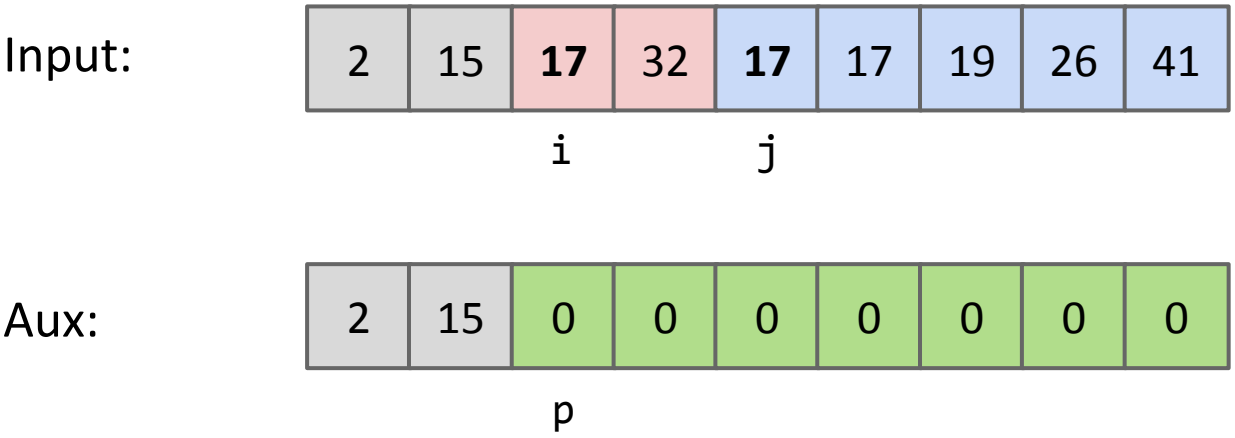
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

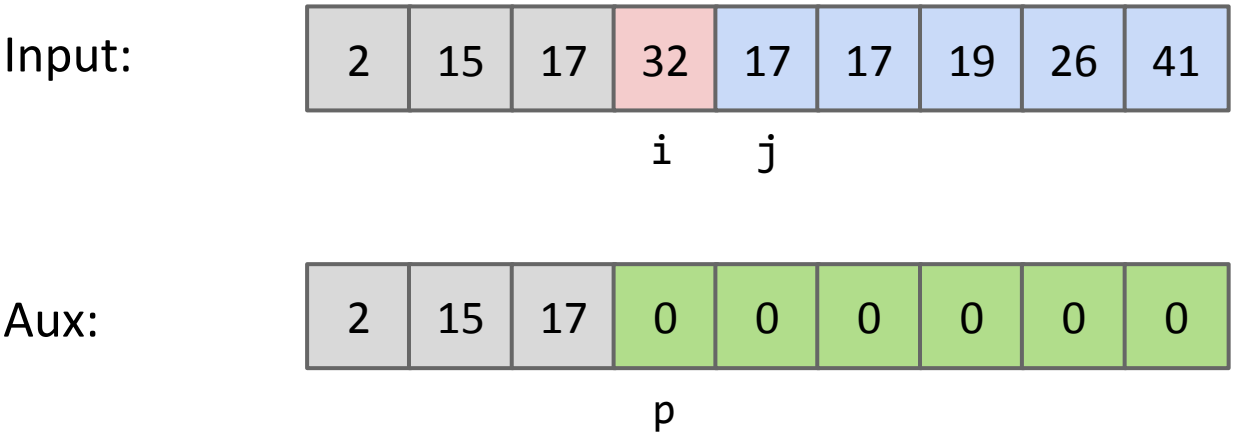
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

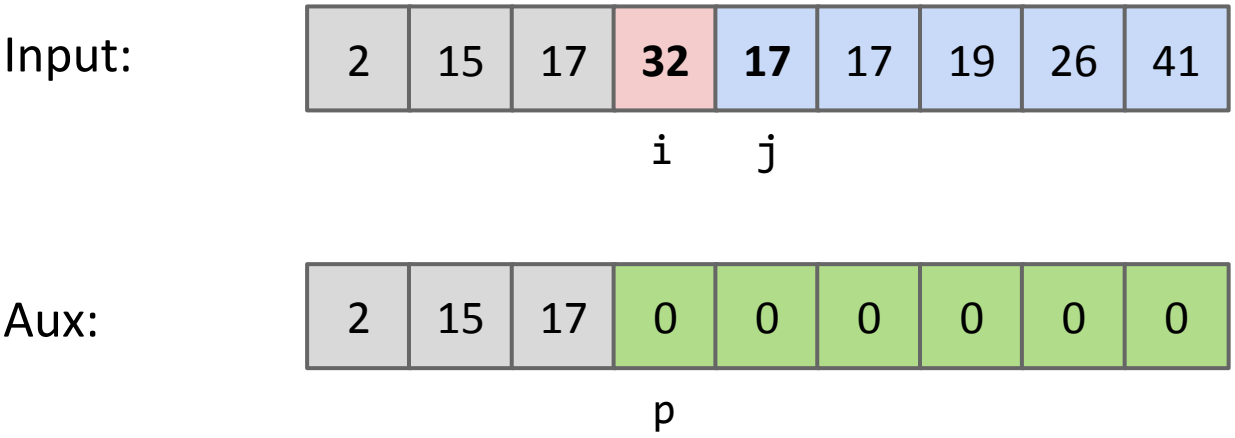
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

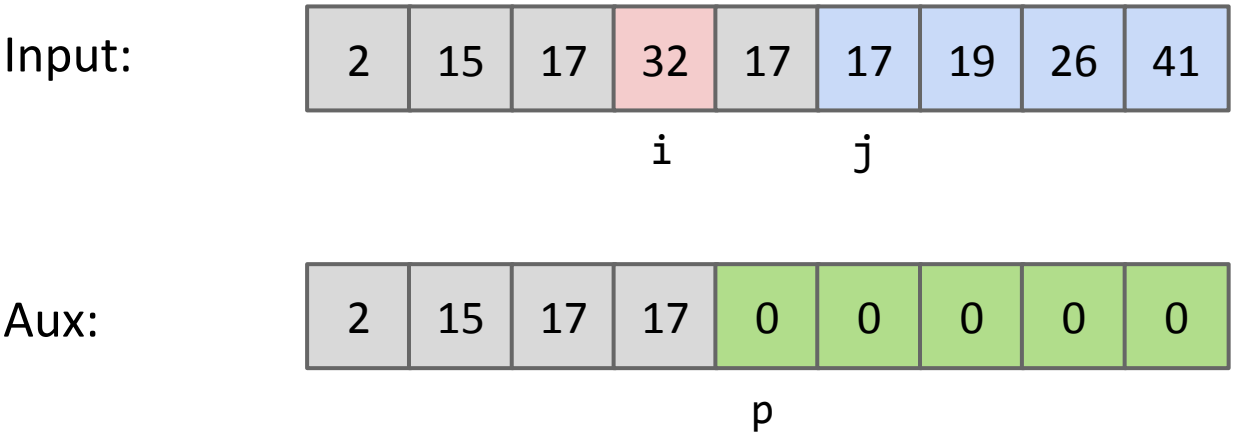
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

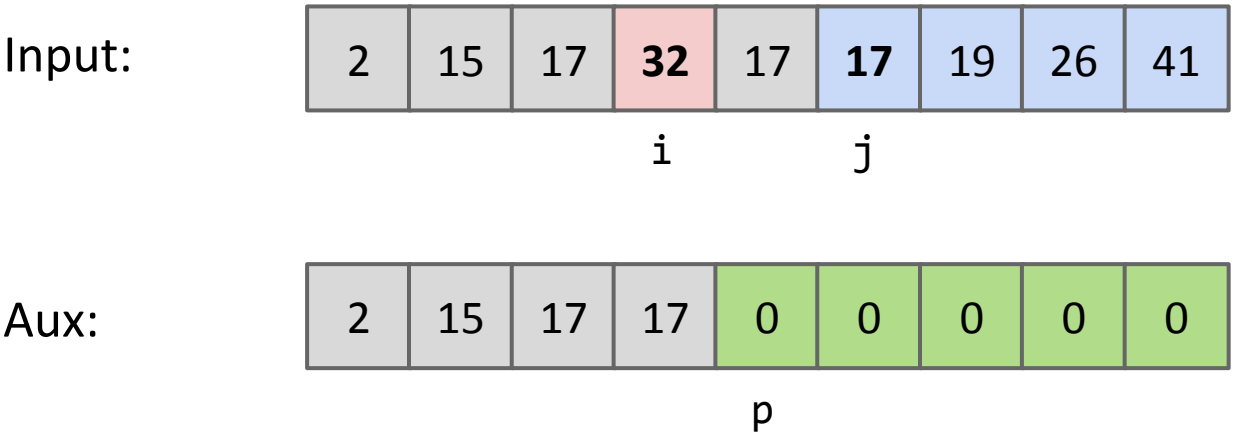
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

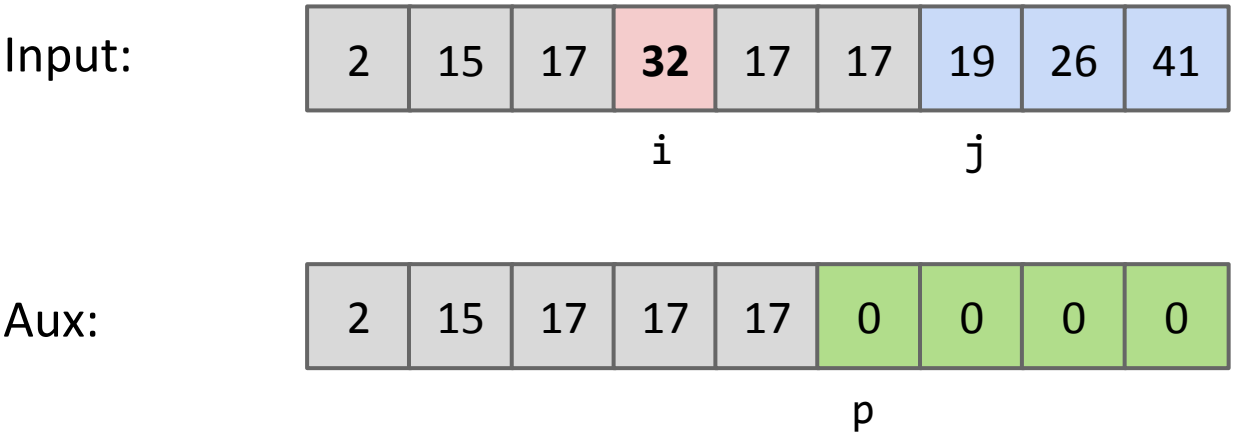
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

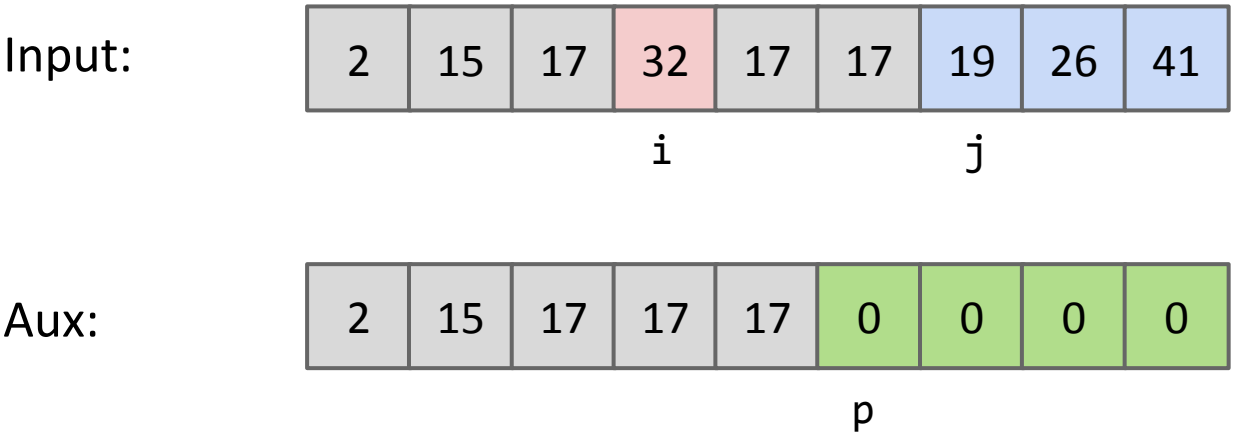
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

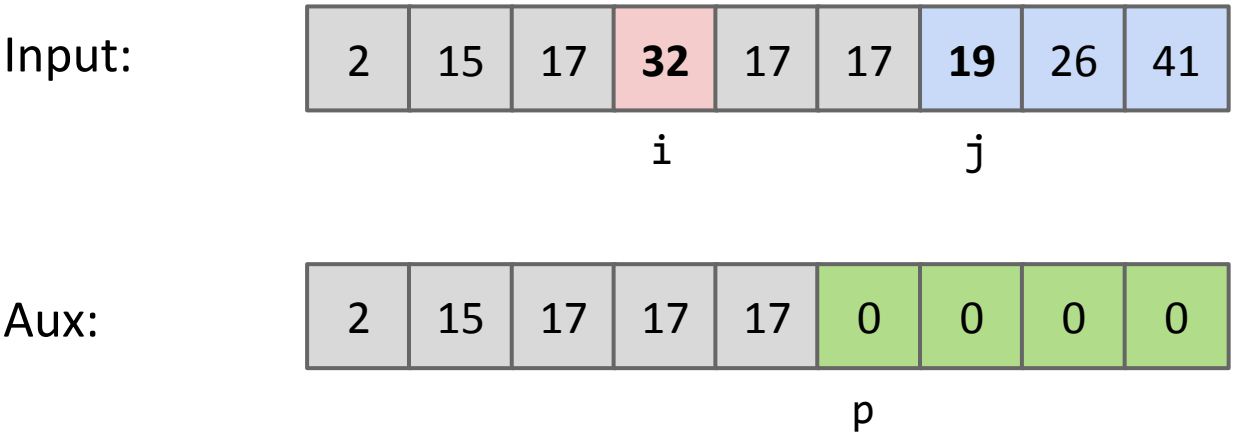
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

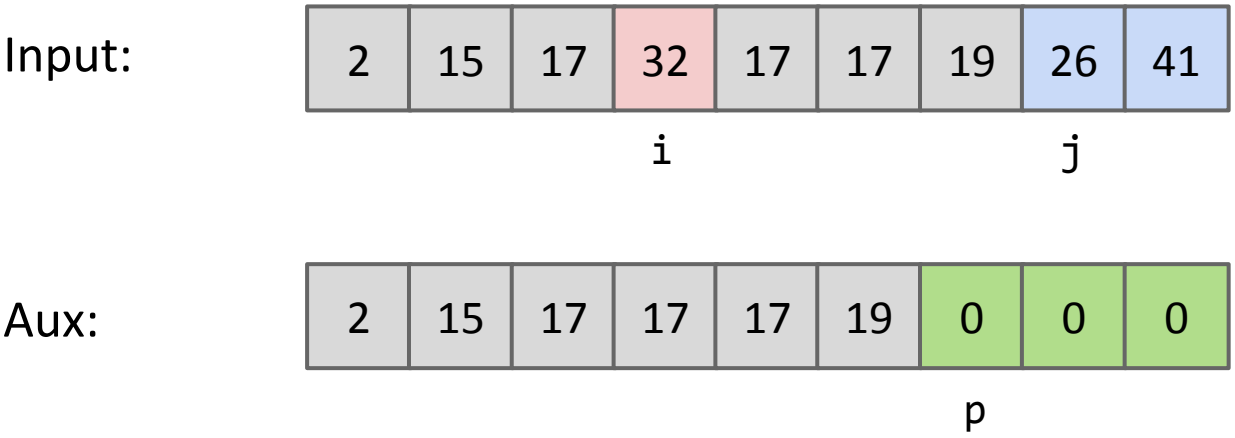
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

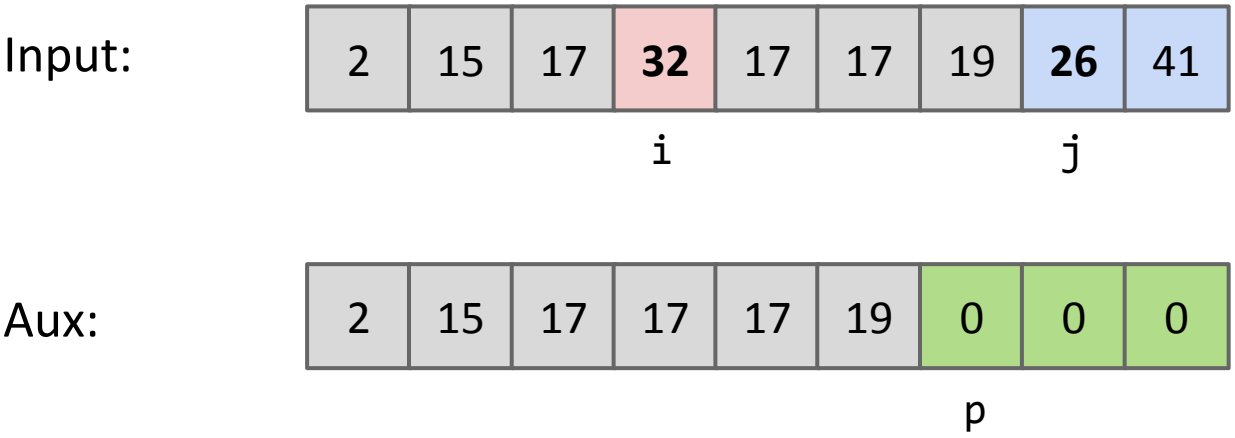
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



Merge Sort

Top-Down merge sorting N items:

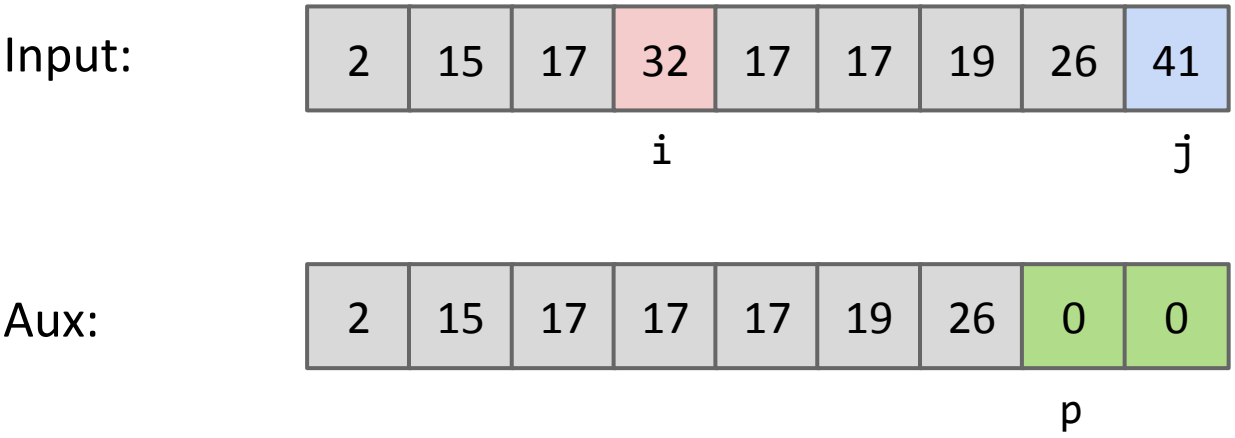
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - **Compare input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.



Merge Sort

Top-Down merge sorting N items:

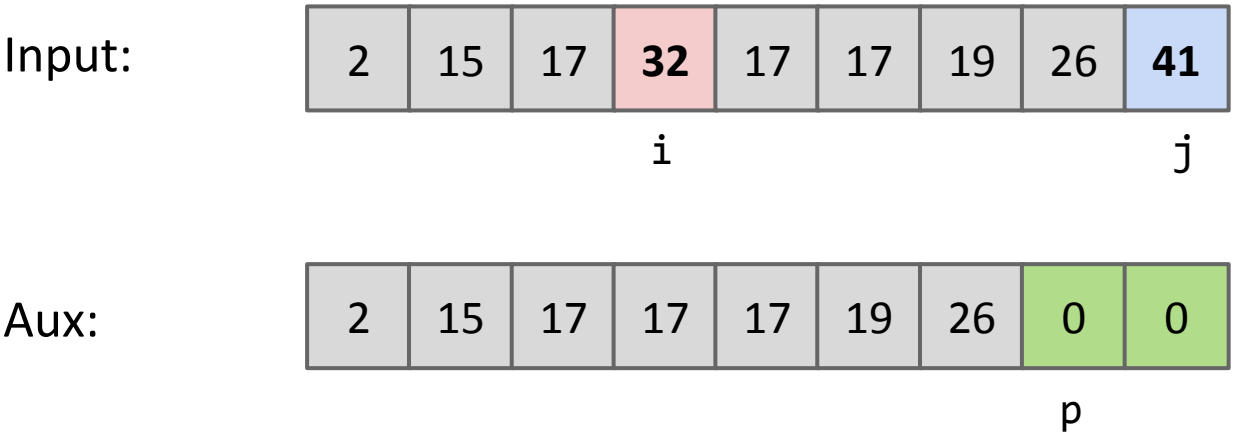
- Split items into 2 roughly even pieces.
- Mergesort each half (steps not shown, this is a recursive algorithm!)
- Merge the two sorted halves to form the final result.
  - Compare input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**



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 input[i] < input[j] (if necessary).**
  - Copy smaller item and increment p and i or j.

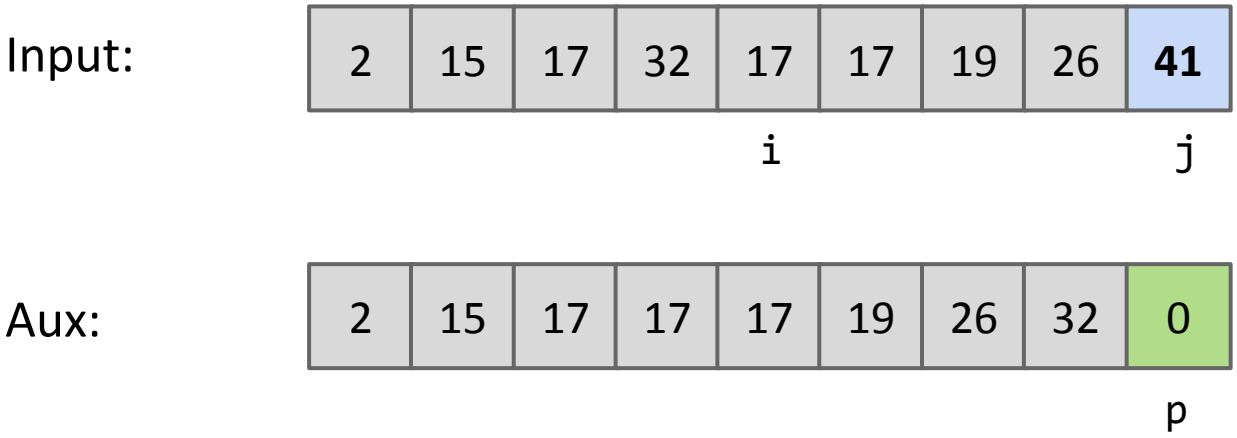


Merge Sort

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 input[i] < 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!



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 input[i] < input[j] (if necessary).
  - **Copy smaller item and increment p and i or j.**