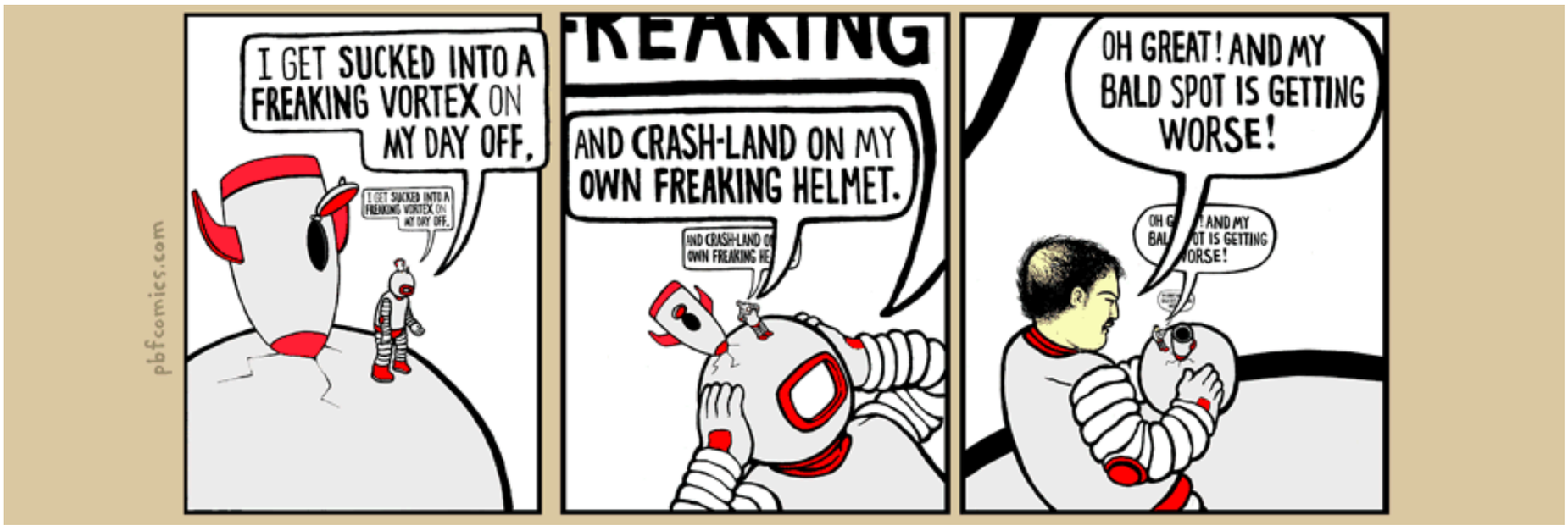
Divide and Conquer: Sorting and Recurrences

Divide & Conquer: The Pattern

- **Divide** the problem into several independent smaller instances of exactly the same problem
- **Delegate** each smaller instance to the **Recursive Leap of Faith** (technically known as induction hypothesis)
- **Combine** the solutions for the smaller instances



Review: Merge Sort

MergeSort(L):

if L has one element
return L

Base case

Divide L into two halves A and B

$A \leftarrow$ **MergeSort**(A)

$B \leftarrow$ **MergeSort**(B)

$L \leftarrow$ **Merge**(A, B)

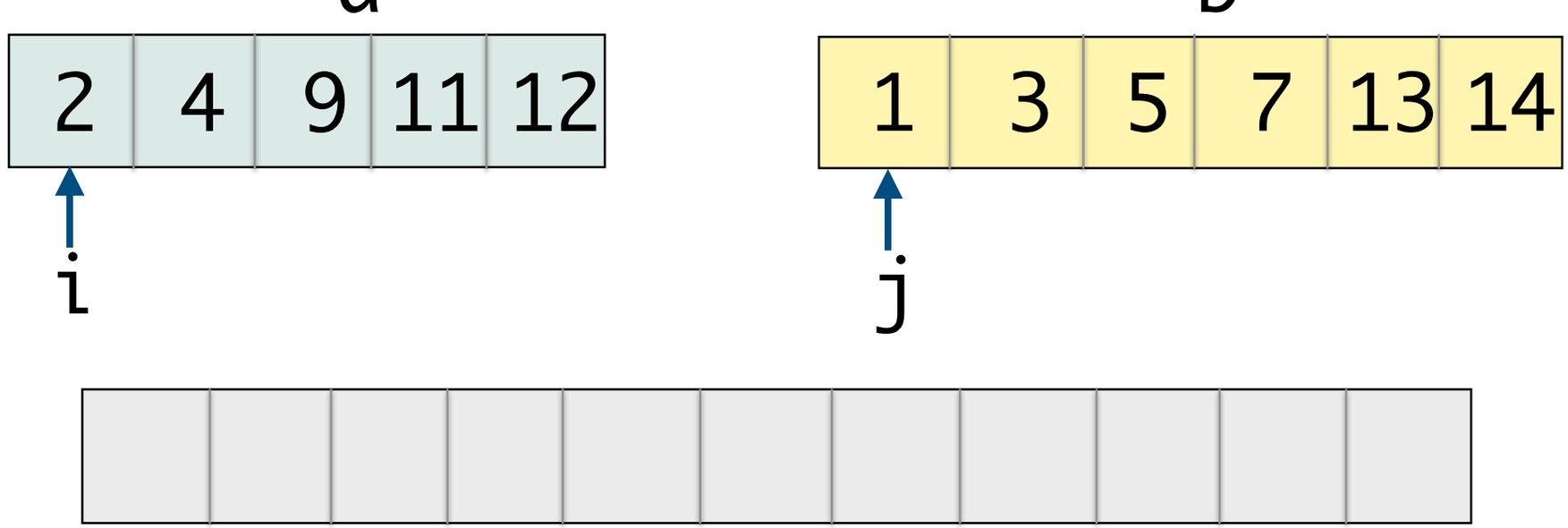
return L

Recursive leaps of faith

Combine solutions

Merge Step: $\Theta(n)$

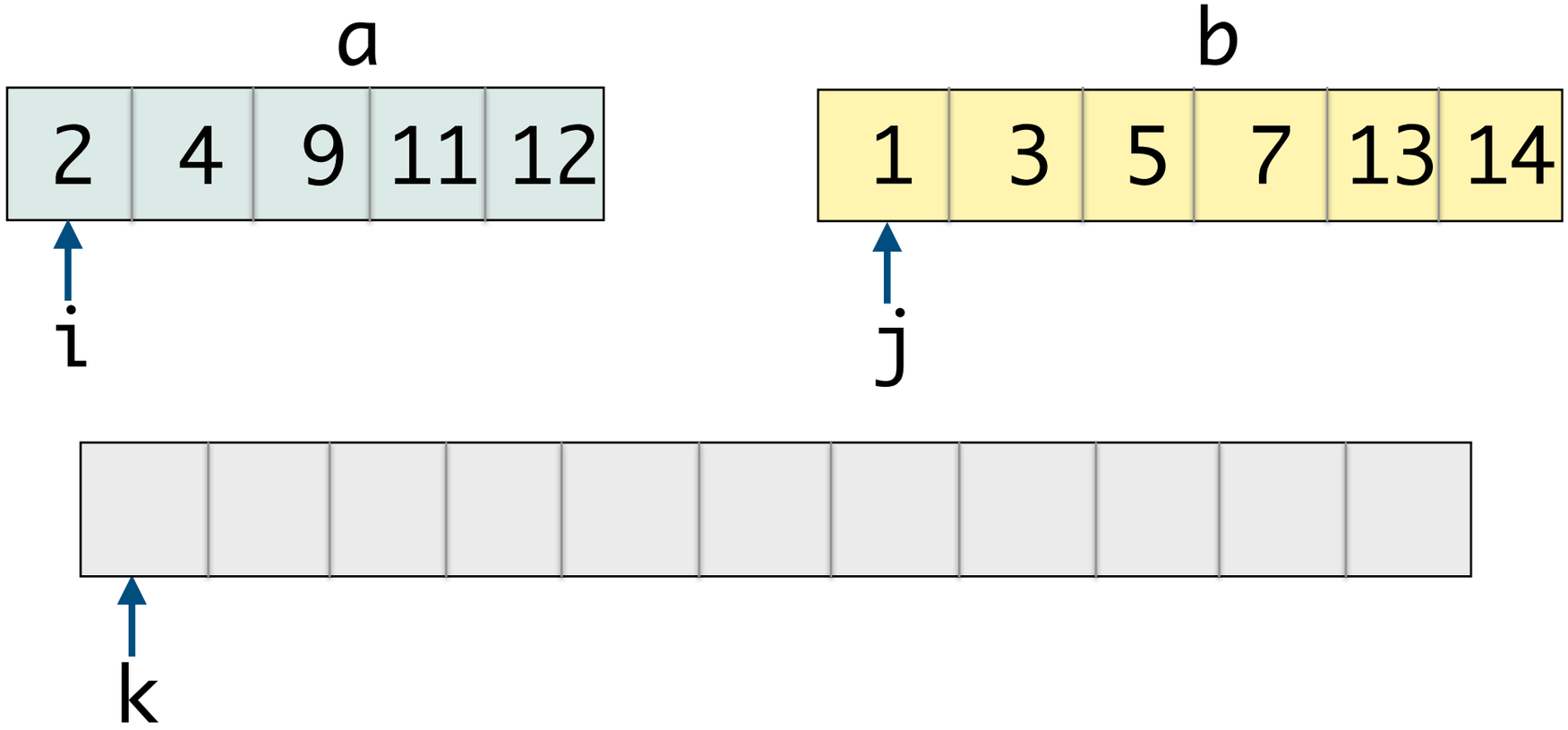
- Scan sorted lists from left to right
- Compare element by element; create new merged list



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

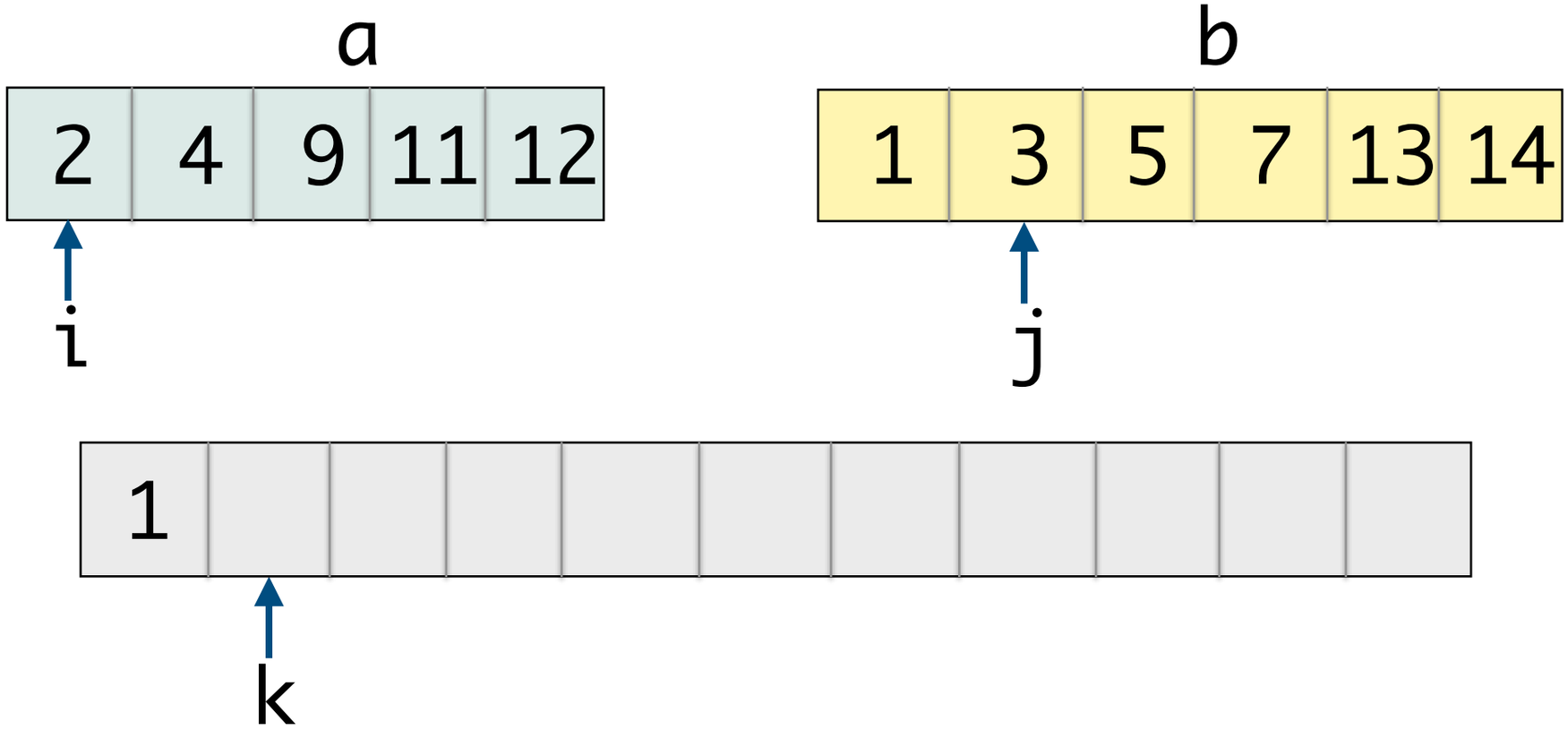
- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

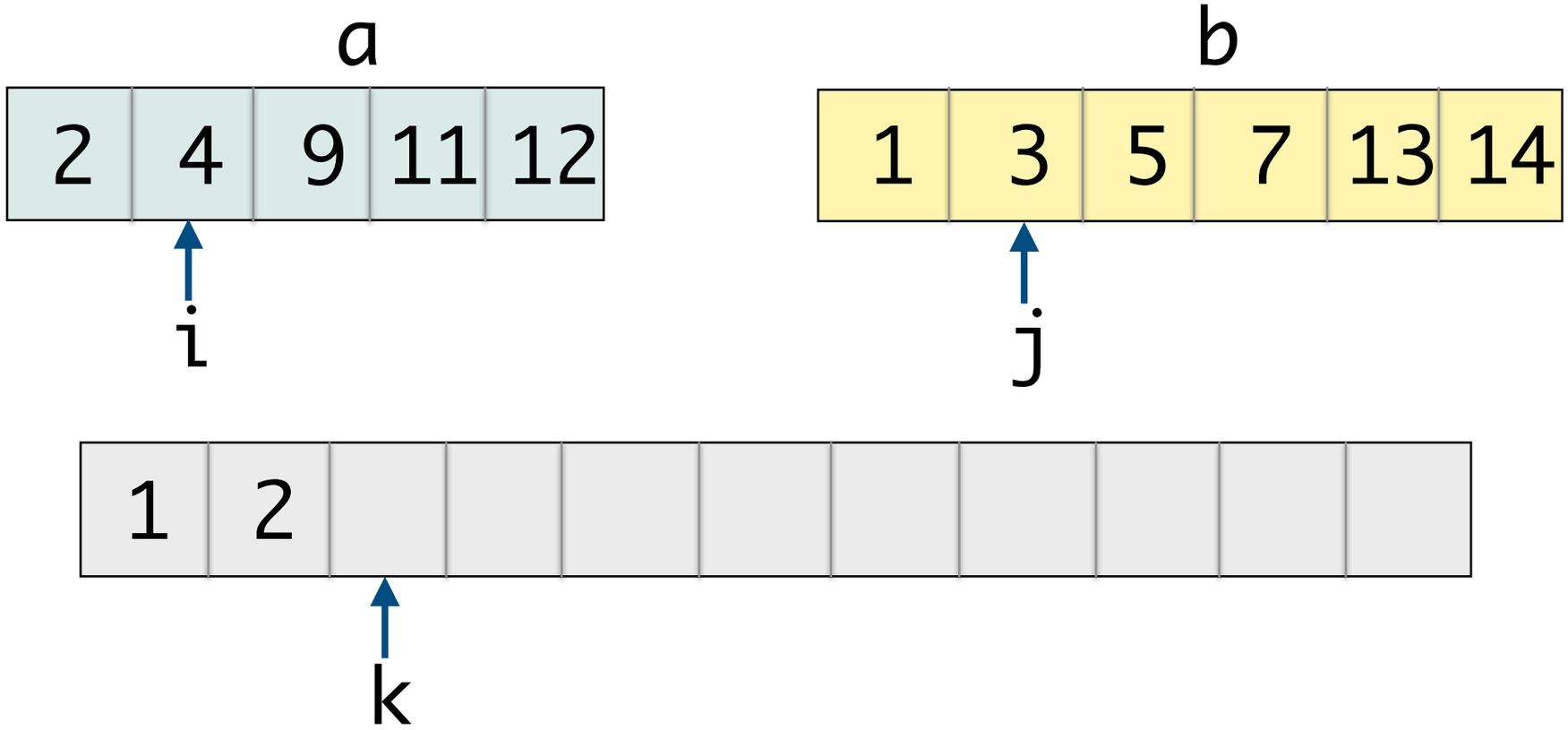
- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

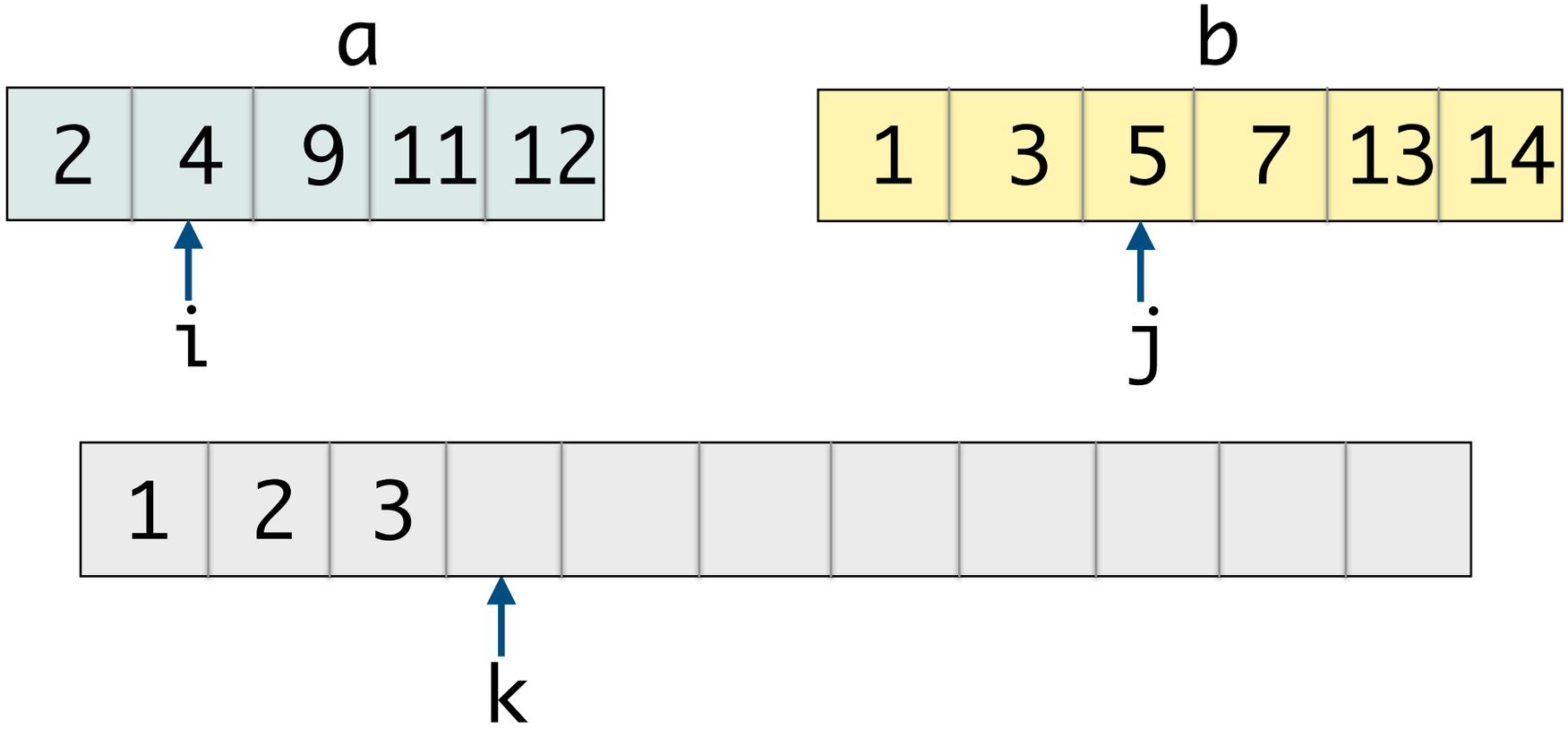
- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

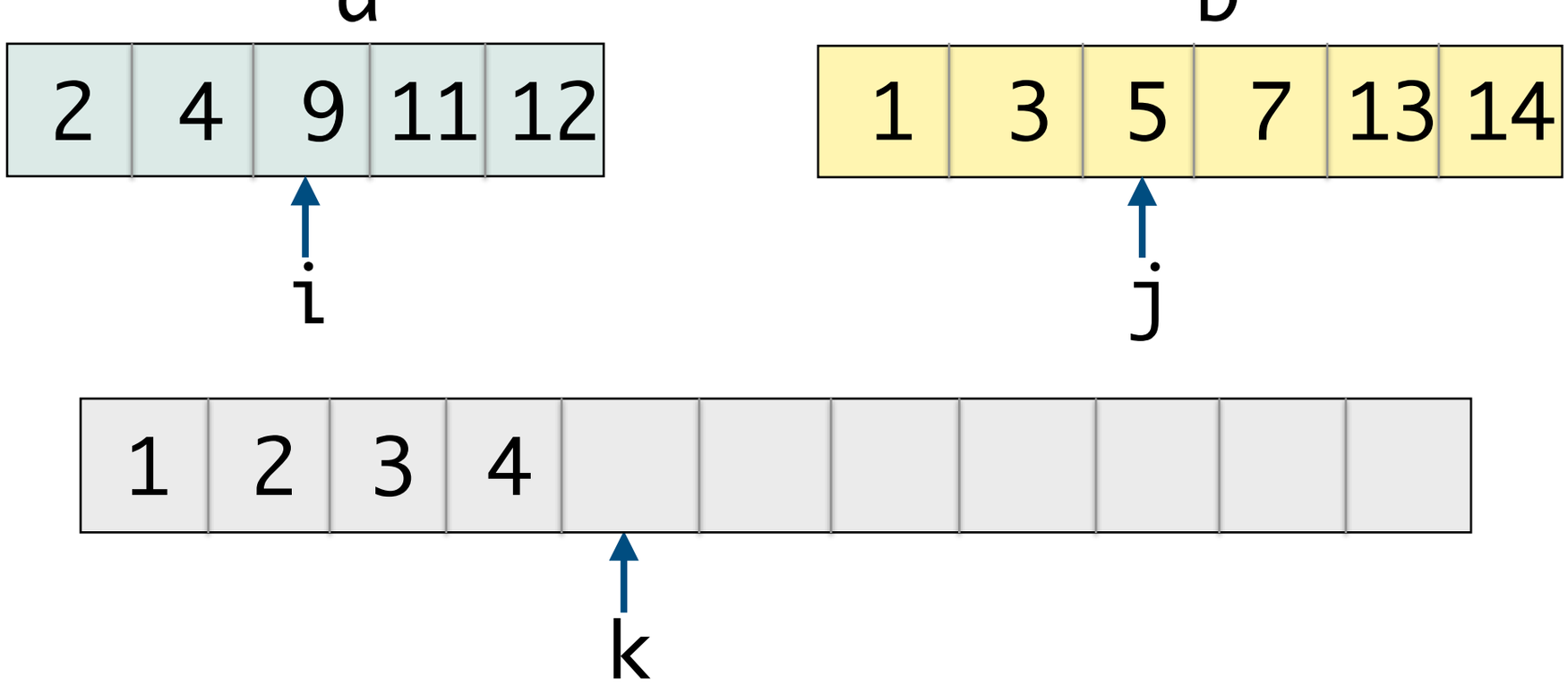
- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

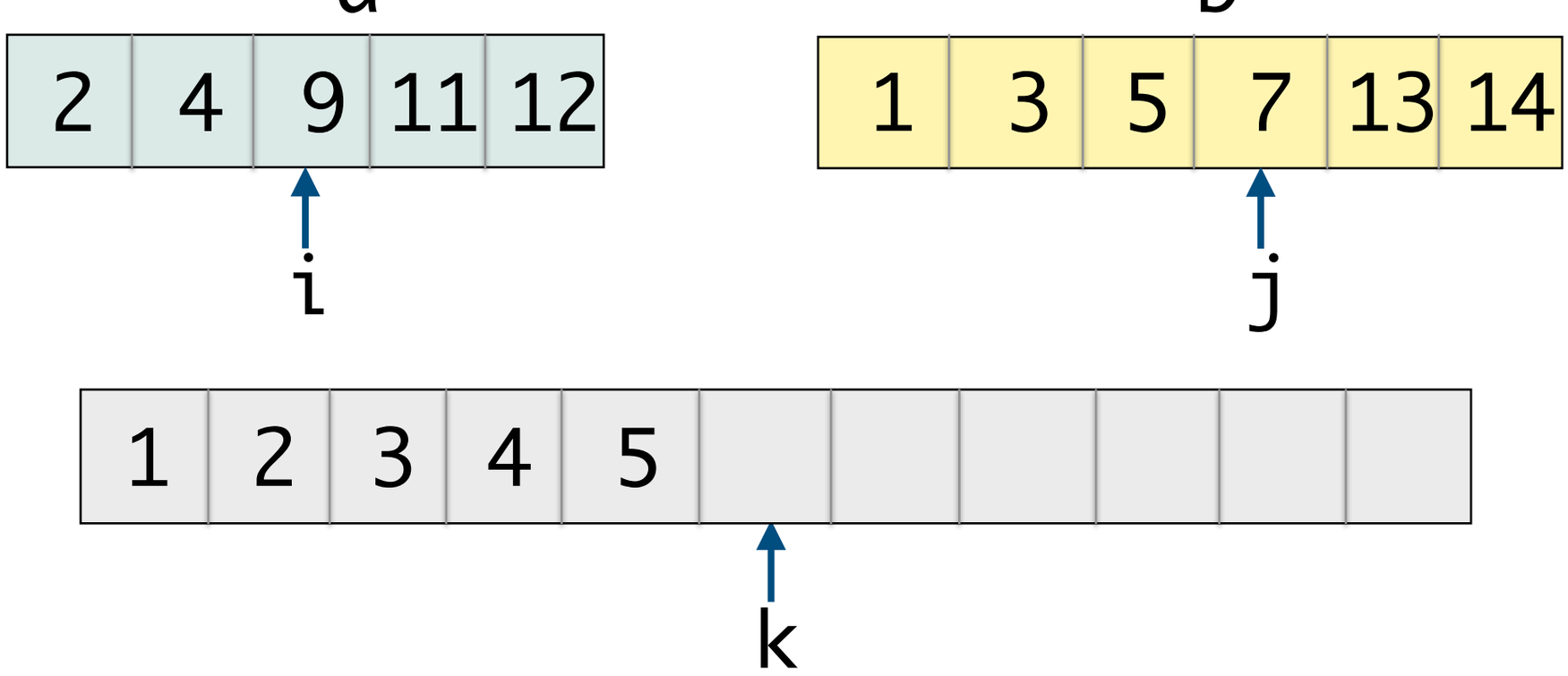
- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j

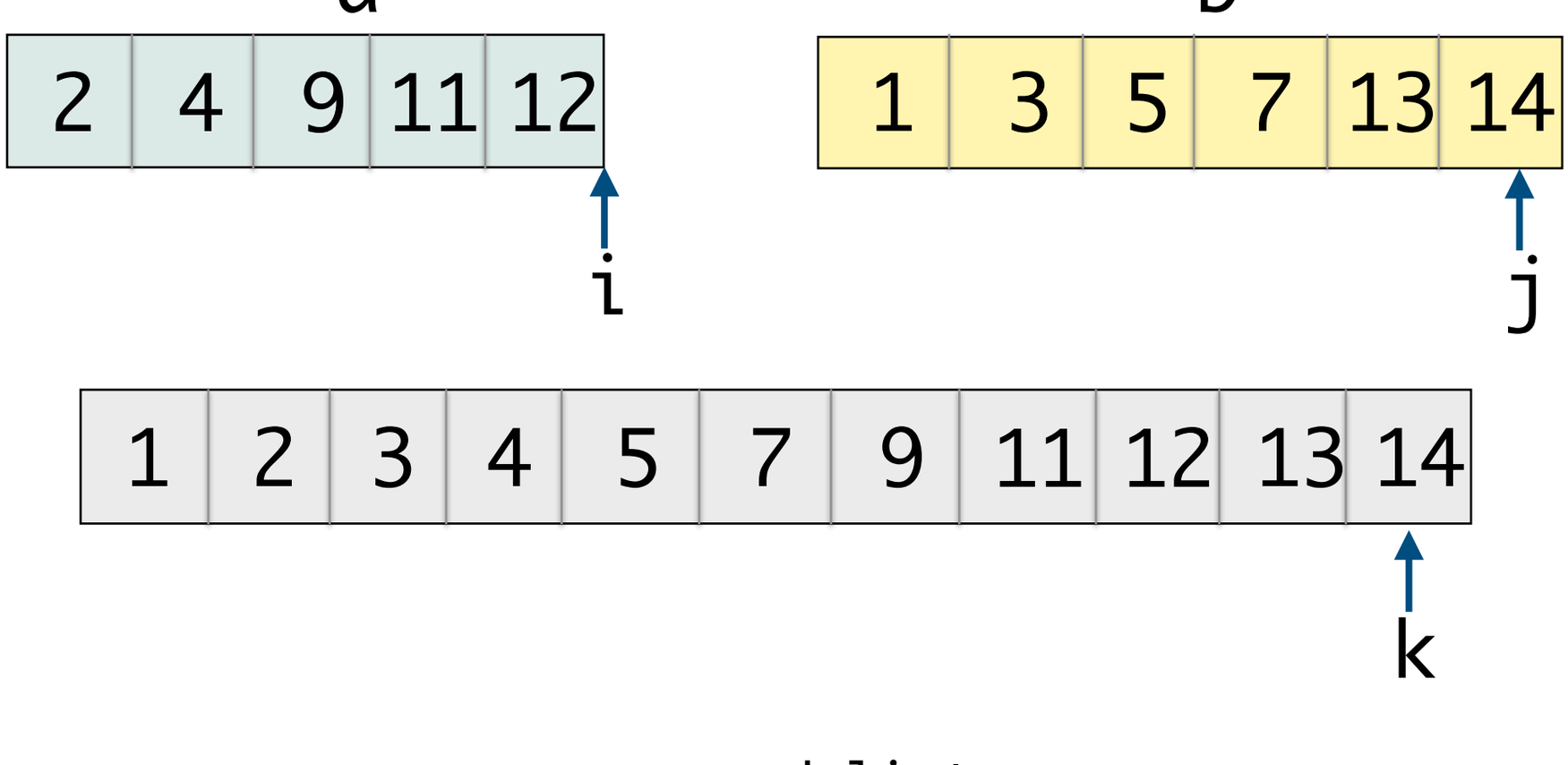


Yada yada yada…

Merge Step: $\Theta(n)$

Is a[i] <= b[j] ?

- Yes, a[i] appended to c, advance i
- No, b[j] appended to c, advance j



Acknowledgments

- Some of the material in these slides are taken from
  - Kleinberg Tardos Slides by Kevin Wayne (https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/04GreedyAlgorithmsI.pdf)
  - Jeff Erickson's Algorithms Book (http://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf)