

2. ANALISI DEGLI ALGORITMI

- ▶ *demo ricerca binaria*
- ▶ *sorted rotated array demo*

Traduzione e adattamento di Vincenzo Bonifaci

Lecture slides by Kevin Wayne

Copyright © 2005 Pearson–Addison Wesley

<http://www.cs.princeton.edu/~wayne/kleinberg-tardos>

Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 33 (esito positivo)

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

↑ ↑
lo hi

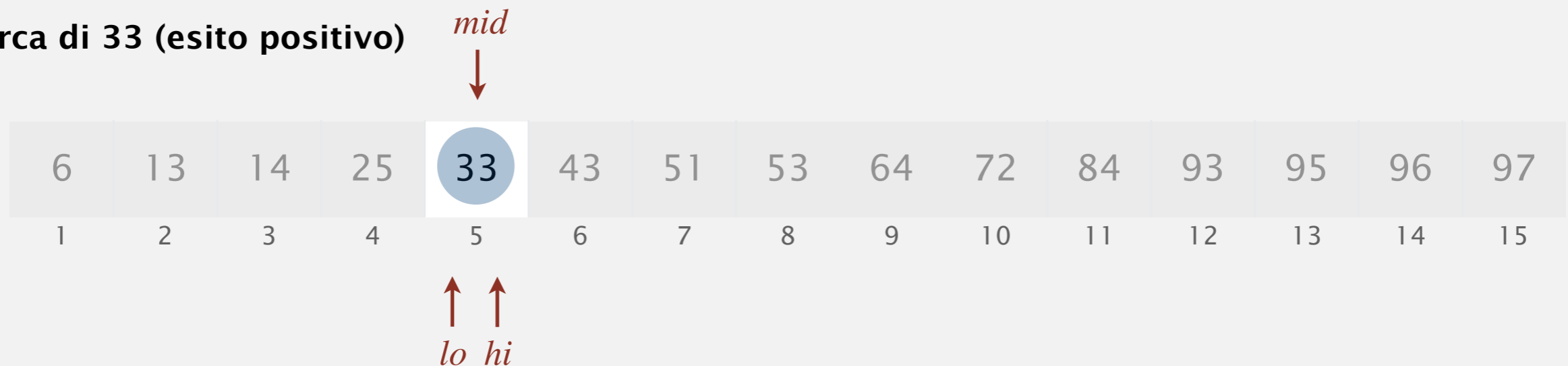
Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 33 (esito positivo)



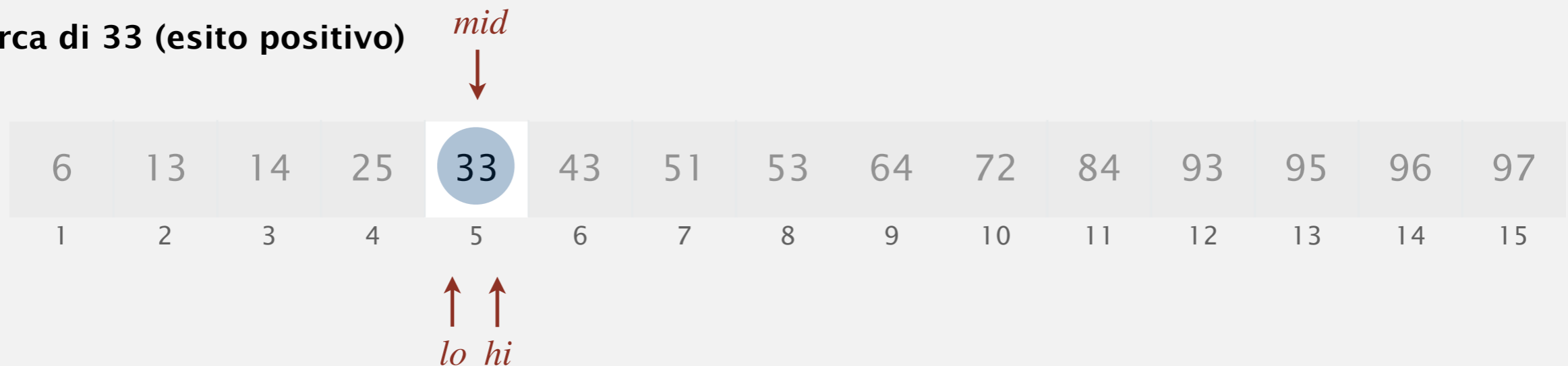
Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 33 (esito positivo)



restituisce indice 5

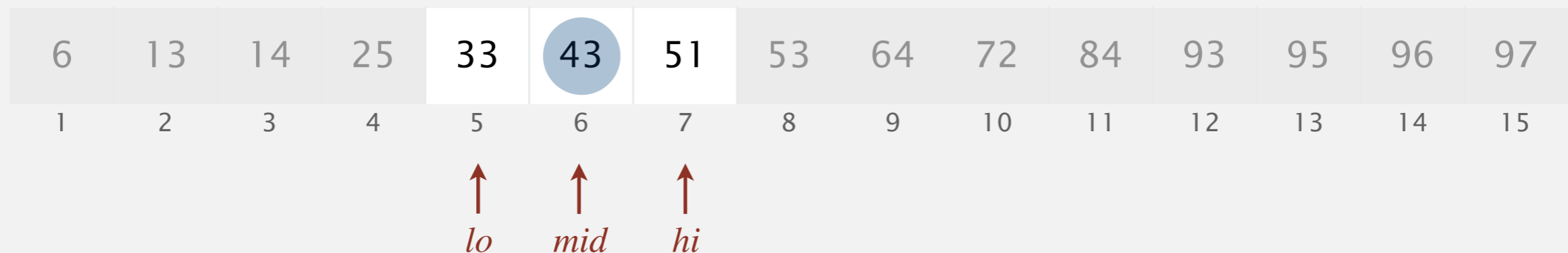
Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 34 (esito negativo)



Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 34 (esito negativo)

6	13	14	25	33	43	51	53	64	72	84	93	95	96	97
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

↑ ↑
lo hi

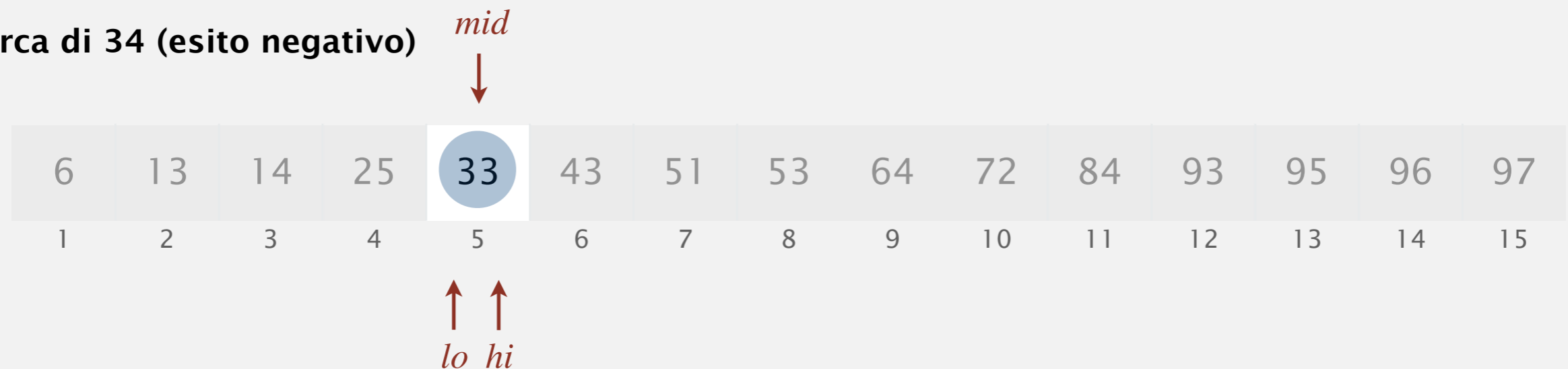
Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 34 (esito negativo)



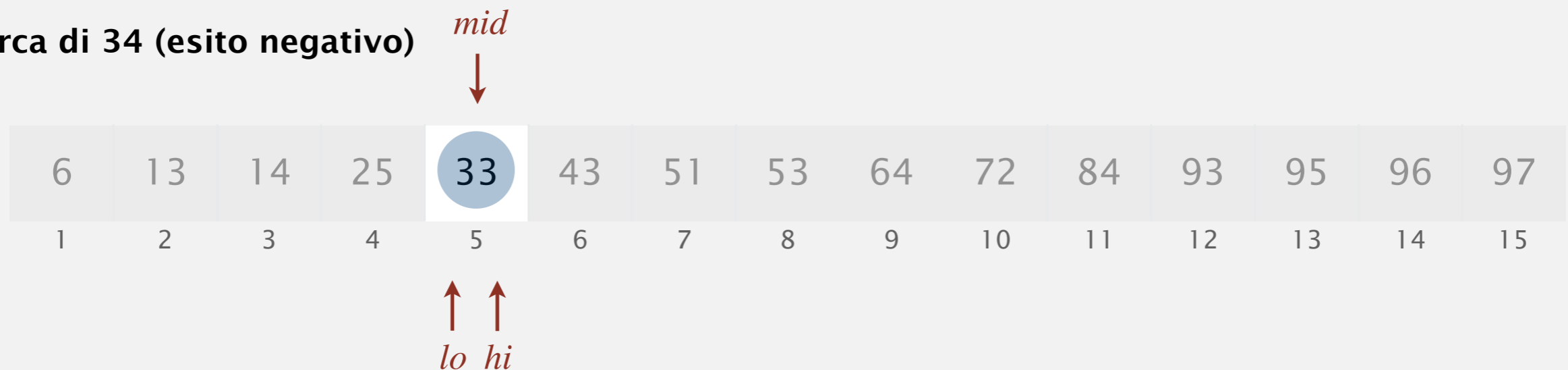
Demo ricerca binaria

Scopo. Dati un array ordinato ed una chiave, trovare l'indice della chiave nell'array.

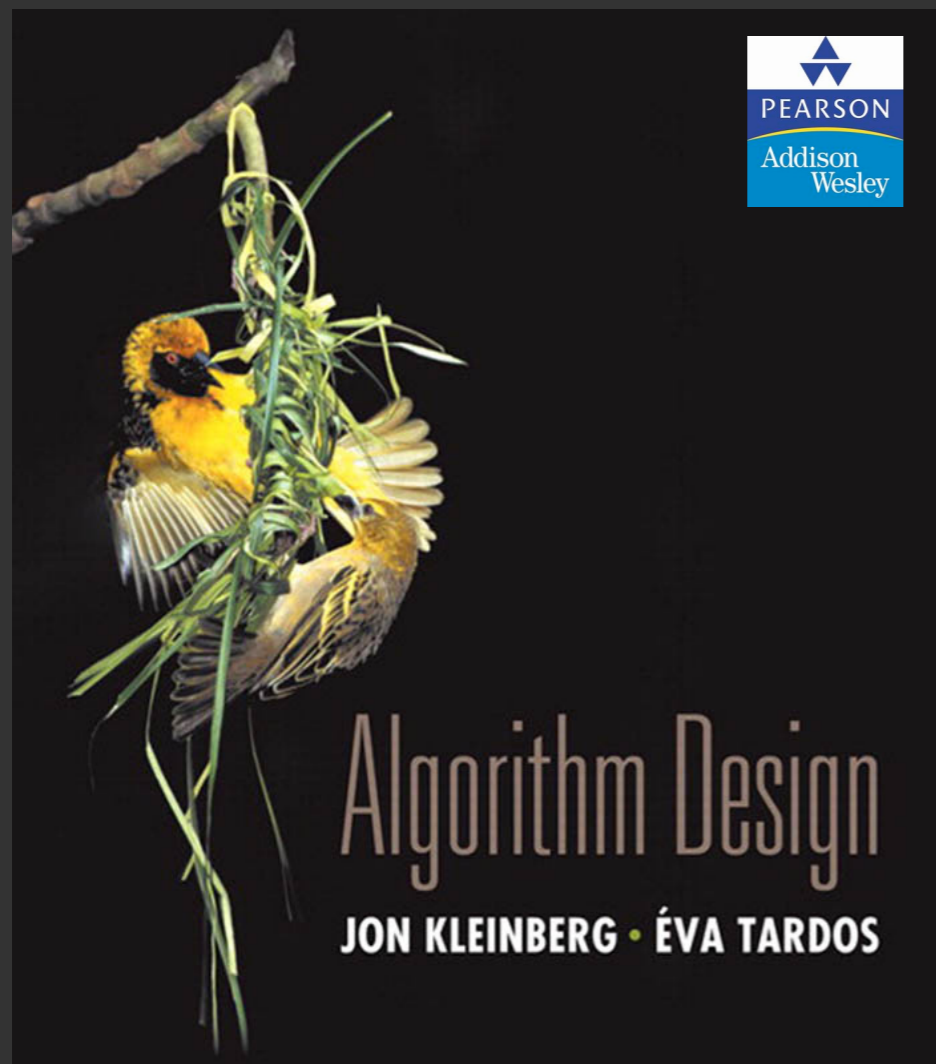
Ricerca binaria. Confronta la chiave con l'elemento di mezzo dell'array.

- Se è più piccola, continua a sinistra.
- Se è più grande, continua a destra.
- Se è uguale, hai trovato l'indice.

ricerca di 34 (esito negativo)



restituisce -1 (non trovato)



2. ANALISI DEGLI ALGORITMI

- ▶ *binary search demo*
- ▶ *demo ricerca in un array ordinato ruotato*

Traduzione e adattamento di Vincenzo Bonifaci

Lecture slides by Kevin Wayne

Copyright © 2005 Pearson–Addison Wesley

<http://www.cs.princeton.edu/~wayne/kleinberg-tardos>

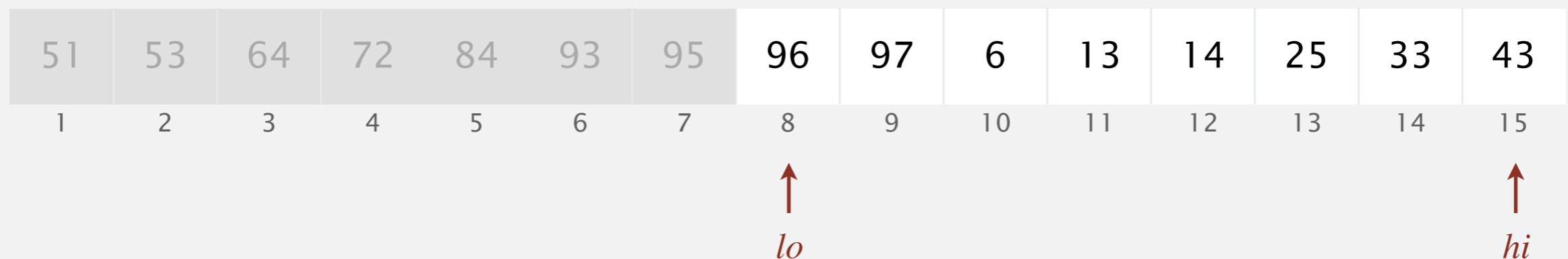
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



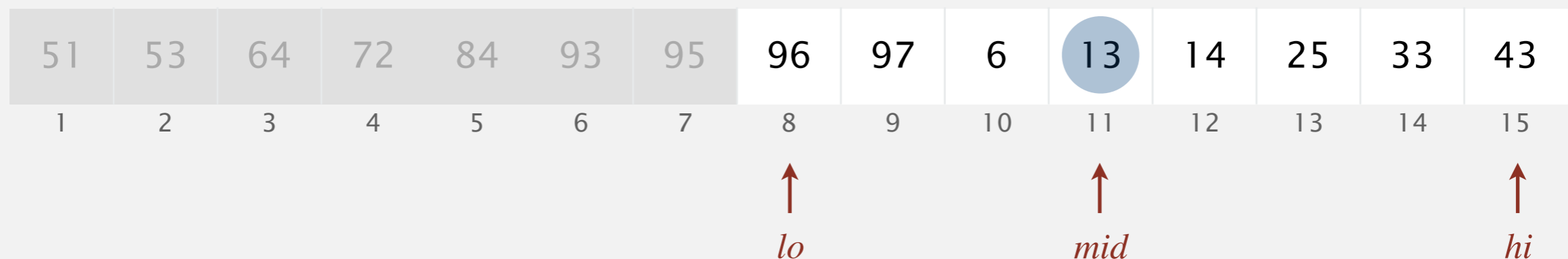
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



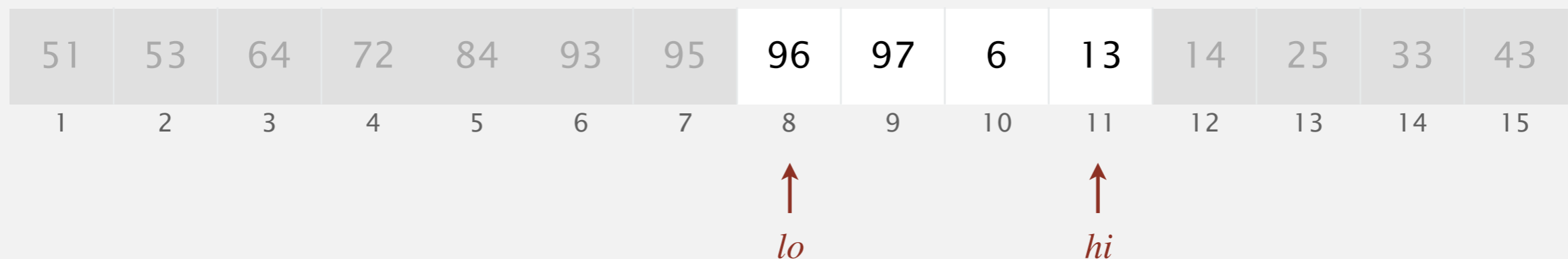
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



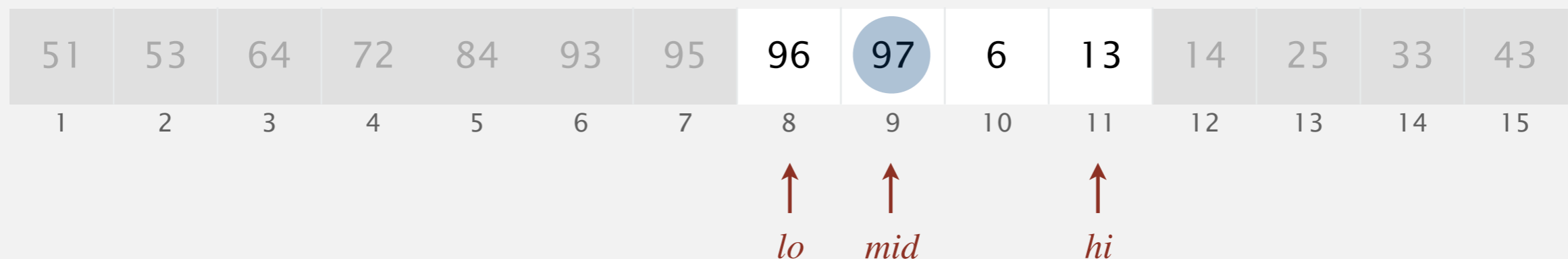
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



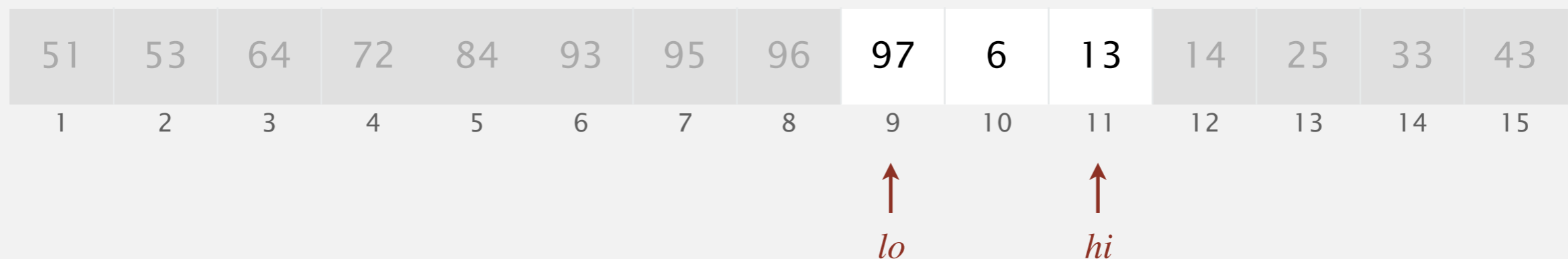
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



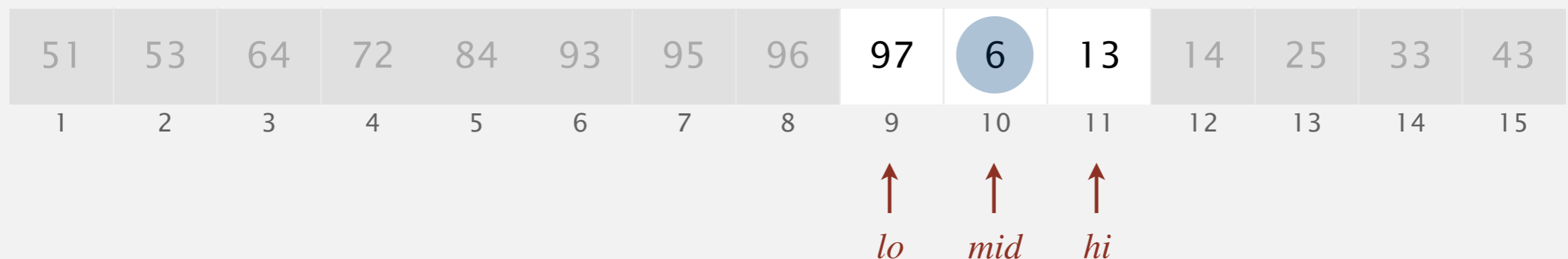
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



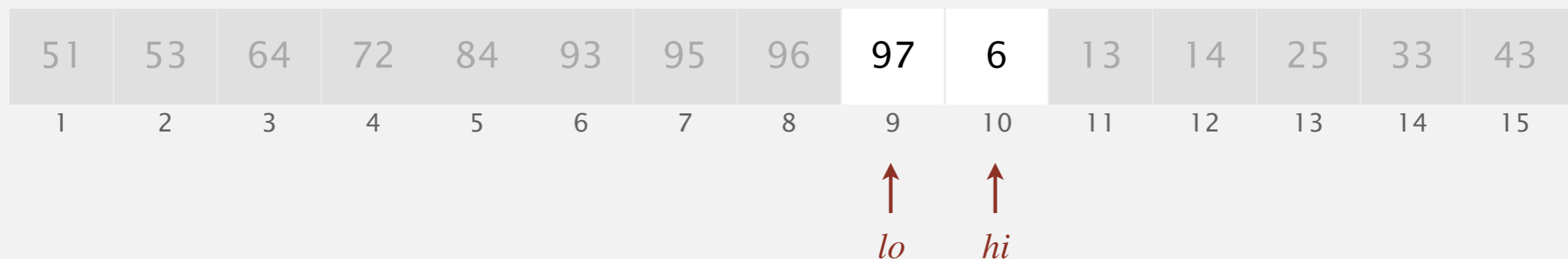
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



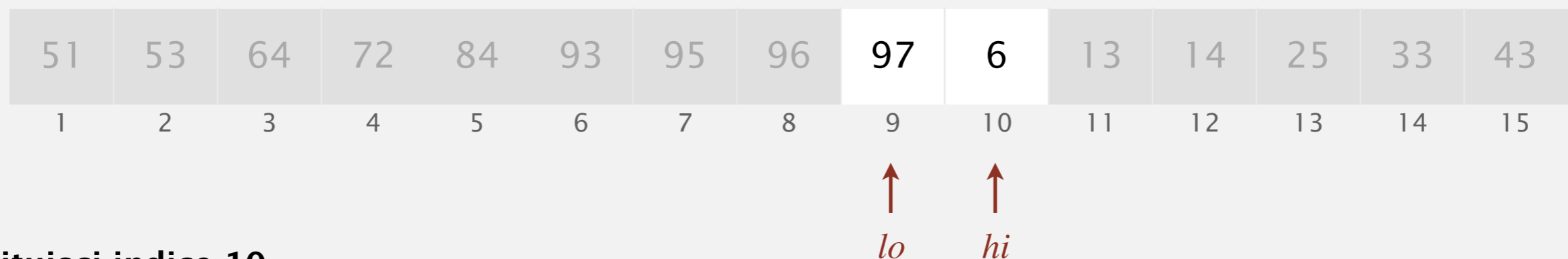
Demo array ordinato ruotato

Scopo. Dato un array ordinato ruotato di n interi distinti, trovare l'indice dell'elemento più piccolo.

Invariante. $A[lo] > A[hi]$.

Ricerca binaria. Confronta l'elemento di mezzo $A[mid]$ con l'ultimo $A[hi]$.

- Se è minore, continua a sinistra.
- Se è maggiore, continua a destra.



restituisce indice 10