

```

/**
 * Searches for an element in a range of indices.
 * @param a the array to search, must be non-null
 * @param lo the (inclusive) low end of the range to search
 * @param hi the (exclusive) high end of the range to search
 * @param x the value to search for
 * ... more tags given below ...
 */
public static int indexOf(int[] a, int lo, int hi, int x)

```

## Specifications

- spec A** \* @requires  $0 \leq lo \leq hi \leq a.length$  and  $x$  occurs in  $a[lo..hi]$   
 \* @returns an index  $i$  such that  $lo \leq i < hi$  and  $a[i] == x$
- spec B** \* @requires  $0 \leq lo \leq hi \leq a.length$   
 \* @returns the *smallest* index  $i$  with  $lo \leq i < hi$  and  $a[i] == x$ ,  
 \* or  $-1$  if no such index exists
- spec C** \* @requires  $0 \leq lo \leq hi \leq a.length$   
 \* @returns the *largest* index  $i$  with  $lo \leq i < hi$  and  $a[i] == x$ ,  
 \* or  $-1$  if no such index exists
- spec D** \* @requires  $0 \leq lo \leq hi \leq a.length$   
 \* @returns *some* index  $i$  with  $lo \leq i < hi$  and  $a[i] == x$ ,  
 \* or  $-1$  if no such index exists
- spec E** \* @requires  $0 \leq lo \leq hi \leq a.length$   
 \* @returns the smallest index  $i$  with  $lo \leq i < hi$  and  $a[i] == x$   
 \* @throws NoSuchElementException if no such index exists

## Implementations

```

impl 1 public static int indexOf(int[] a, int lo, int hi, int x) {
    for (int i = lo; i < hi; i++) {
        if (a[i] == x) {
            return i;
        }
    }
    return -1;
}

impl 2 public static int indexOf(int[] a, int lo, int hi, int x) {
    for (int i = lo; i < hi; i++) {
        if (a[i] == x) {
            return i;
        }
    }
    throw new NoSuchElementException("not found: " + x);
}

impl 3 public static int indexOf(int[] a, int lo, int hi, int x) {
    int i = lo;
    while (a[i] != x) {
        i++;
    }
    return i;
}

impl 4 public static int indexOf(int[] a, int lo, int hi, int x) {
    for (int i = hi - 1; i >= lo; i--) {
        if (a[i] == x) {
            return i;
        }
    }
    return -1;
}

```