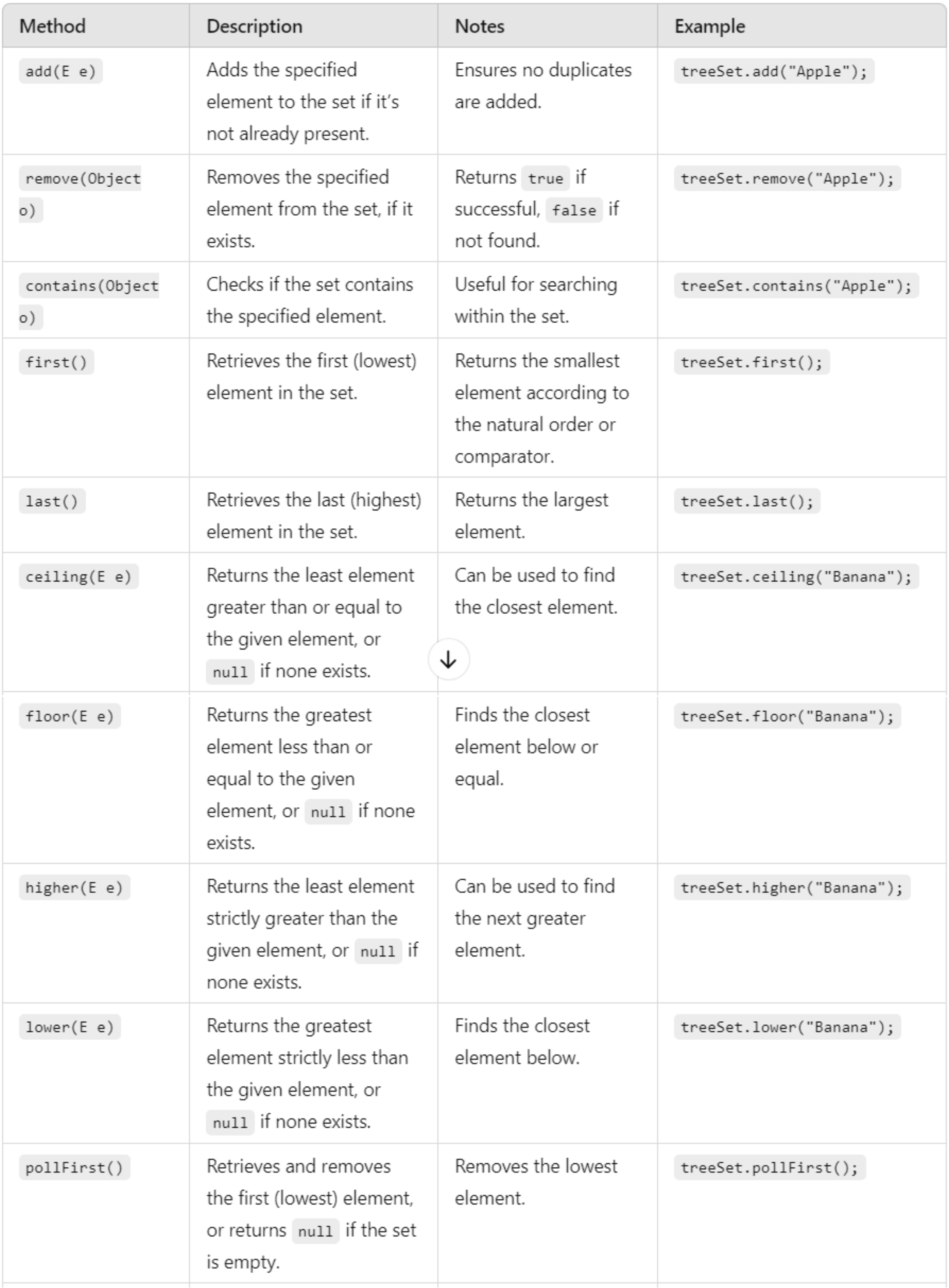
TreeSet Methods Overview and Examples:



Method	Description	Notes	Example
`add(E e)`	Adds the specified element to the set if it's not already present.	Ensures no duplicates are added.	`treeSet.add("Apple");`
`remove(Object o)`	Removes the specified element from the set, if it exists.	Returns `true` if successful, `false` if not found.	`treeSet.remove("Apple");`
`contains(Object o)`	Checks if the set contains the specified element.	Useful for searching within the set.	`treeSet.contains("Apple");`
`first()`	Retrieves the first (lowest) element in the set.	Returns the smallest element according to the natural order or comparator.	`treeSet.first();`
`last()`	Retrieves the last (highest) element in the set.	Returns the largest element.	`treeSet.last();`
`ceiling(E e)`	Returns the least element greater than or equal to the given element, or `null` if none exists.	Can be used to find the closest element.	`treeSet.ceiling("Banana");`
`floor(E e)`	Returns the greatest element less than or equal to the given element, or `null` if none exists.	Finds the closest element below or equal.	`treeSet.floor("Banana");`
`higher(E e)`	Returns the least element strictly greater than the given element, or `null` if none exists.	Can be used to find the next greater element.	`treeSet.higher("Banana");`
`lower(E e)`	Returns the greatest element strictly less than the given element, or `null` if none exists.	Finds the closest element below.	`treeSet.lower("Banana");`
`pollFirst()`	Retrieves and removes the first (lowest) element, or returns `null` if the set is empty.	Removes the lowest element.	`treeSet.pollFirst();`

`pollLast()`	Retrieves and removes the last (highest) element, or returns `null` if the set is empty.	Removes the highest element.	`treeSet.pollLast();`
`subSet(E from, E to)`	Returns a view of the portion of this set within the specified range (`from` inclusive, `to` exclusive).	Allows viewing a range of elements.	`treeSet.subSet("Apple", "Mango");`
`headSet(E to)`	Returns a view of the portion of this set strictly less than the specified element.	Useful for range filtering.	`treeSet.headSet("Mango");`
`tailSet(E from)`	Returns a view of the portion of this set greater than or equal to the specified element.	Useful for range filtering.	`treeSet.tailSet("Banana");`
`descendingSet()`	Returns a view of the set in descending order. This is the reverse order of the `TreeSet` .	Iterates through the set in reverse order.	`treeSet.descendingSet();`