Given an array of integers `a`, return a new array `b` using the following guidelines:

- For each index `i` in `b`, the value of b_i is the index of the a_j nearest to a_i and is also greater than a_i.
- If there are two options for b_i, put the leftmost one in b_i.
- If there are no options for b_i, put `-1` in b_i.

Example

For `a = [1, 4, 2, 1, 7, 6]`, the output should be
`nearestGreater(a) = [1, 4, 1, 2, -1, 4]`.

- for `a[0]`, the nearest larger element is `4` at index `a[1]` -> `b[0]` contains the value `1`.
- for `a[1]`, the nearest larger element is `7` at `a[4]` -> `b[1]` contains the value `4`.
- for `a[2]`, the nearest larger element is `4` at `a[1]` (`7`is also larger, but `4` has the minimal position) -> `b[2]` contains the value `1`.
- for `a[3]`, the nearest larger element is `2` at `a[2]` (`7`is also larger, but `2` has the minimal position) -> `b[3]` contains the value `2`.
- for `a[4]`, there is no element larger than `7` -> `b[4]` contains the value `-1`.
- for `a[5]`, the nearest larger element is `7` at `a[4]` -> `b[5]` contains the value `4`.

Input/Output

- **[time limit] 4000ms (py3)**

- **[input] array.integer a**

  An unsorted array of integers.

  Guaranteed constraints:
  $1 \le a.length \le 10^4$,
  $1 \le a[i] \le 10^9$.

- **[output] array.integer**

  An array `b`, where for each index `i`, b_i is the index of the nearest number in `a`that is greater than a_i.