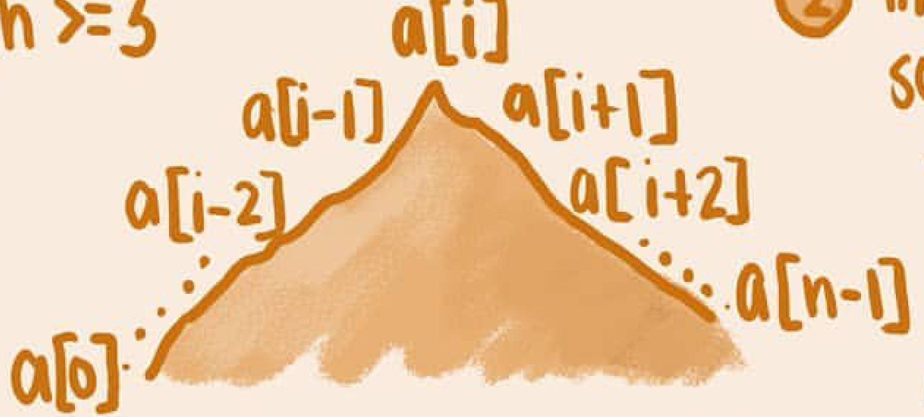
852. Peak Index in a Mountain Array

An array 'a' is called a mountain if-

1. A.length >= 3



2. There exists some 'i' such that $0 \leq i < n$ and

$a[0] < \ldots < a[i-1] < a[i] > a[i+1] > \ldots > a[n-1]$

Find and return peak index 'i' given it always exist.

ex. Input: [0,1,0] → Output : 1

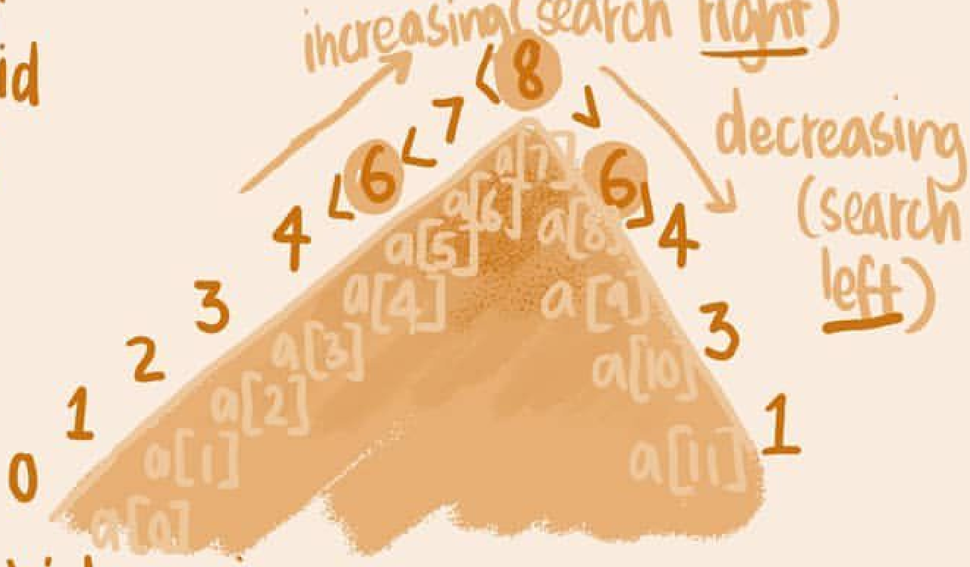
the code dose

852. Peak Index in a Mountain Array

APPROACH - USING BINARY SEARCH

A = [0, 1, 2, 3, 4, 6, 7, 8, 6, 4, 3, 1]

mid



```
① Set low = 0
      high = len(A)-1

② Repeat while
   low < high

   (i) find mid = $\frac{low + high}{2}$
                          (increasing)
   (ii) if a[mid-1] < a[mid] < a[mid+1] search right
                          (decreasing)
   (iii) if a[mid-1] > a[mid] > a[mid+1] search left
   (iv) if a[mid-1] < a[mid] > a[mid+1] return mid
                                                  peak index
```

852. Peak Index in a Mountain Array

```
def peakIndexInMountainArray(self, A: List[int]) -> int:
    low = 0
    high = len(A) - 1

    while low < high:
        mid = (low + high) >> 1
        ele = A[mid]
        after = A[mid + 1]
        before = A[mid - 1]

        if before < ele and ele < after:
            low = mid + 1  search right
        elif before < ele and ele > after:
            return mid  found peak
        elif before > ele and ele > after:
            high = mid  search left
```

the code dose