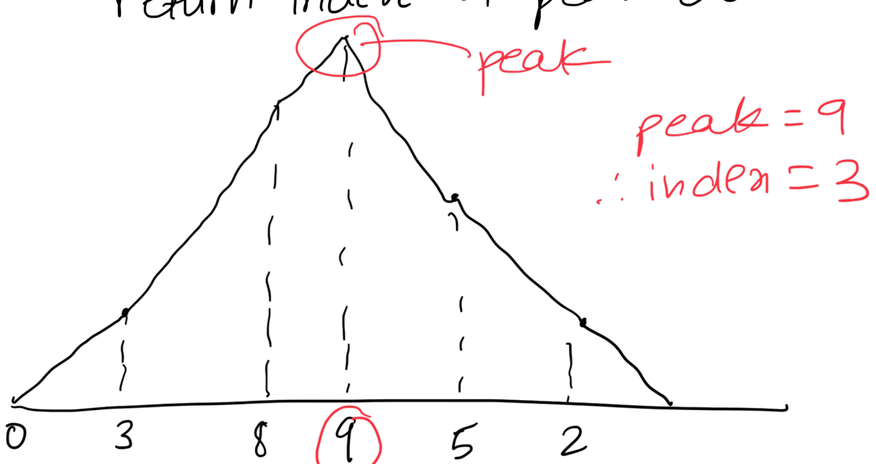
# Peak index in mountain array

arr = [0, 3, 8, 9, 5, 2]

return index of peak element



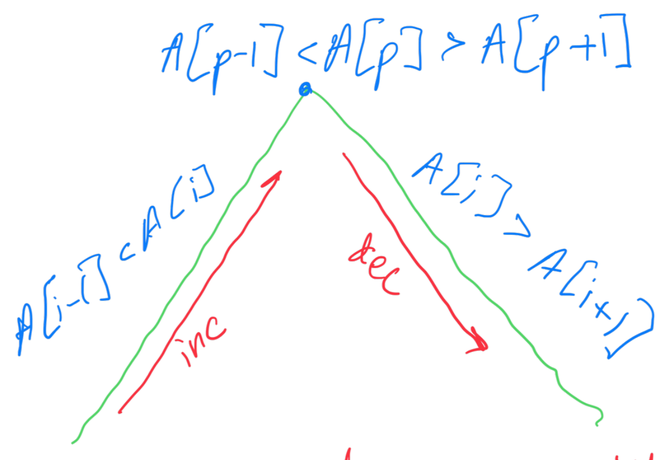

peak = 9

∴ index = 3

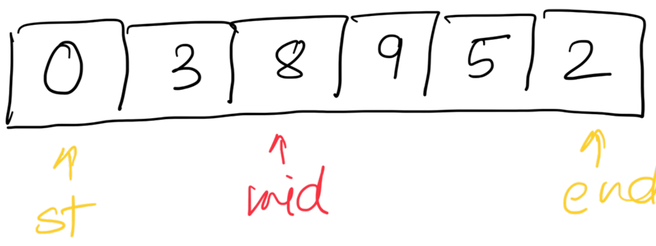
$A[i-1] < A[i]$

$A[i] > A[i+1]$

$A[p-1] < A[p] > A[p+1]$




and also its always a mountain array so answer will never be at idx 0 & n-1 (so exclude from search space)




① $mid = s + \frac{(e-s)}{2}$

② $A[mid-1] < A[mid] > A[mid+1]$

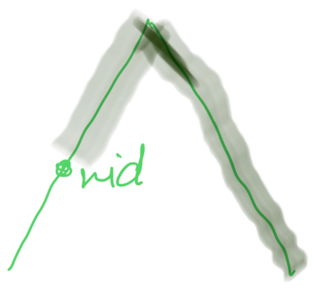
return mid

③ Search space

L ↙ ↘ R ??

Case 1 : mid = peak

Case 2 :




peak will always lie on the right side

Case 3 :


peak will always lie on the left side

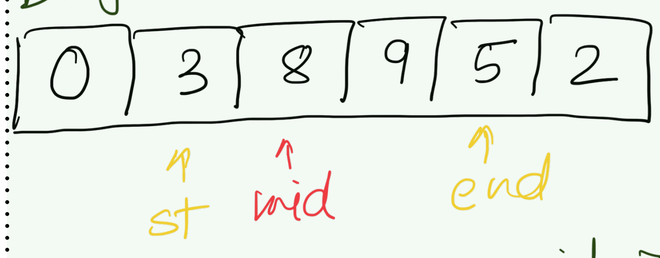
mid

↙ lies on the left ↘ lies on the right

$A[mid-1] < A[mid]$ ⇒ st = mid + 1 ↓ do right search

else ⇒ end = mid - 1 ↓ do left search

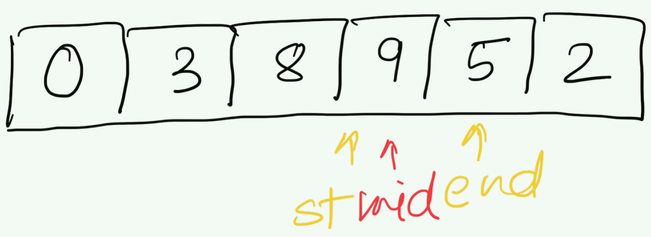
Dry run




mid > st but mid ≠ end ⇒ not peak

mid > st ⇒ search space is on the right side

st = mid + 1




mid > st and mid > end ⇒ PEAK

Pseudo code

```
st = 1, end = n-2
while (st <= end) {
    mid = st + (end-st)/2
    if (A[mid-1] < A[mid] > A[mid+1]) {
        return mid;
    }
    if (A[mid-1] < A[mid]) {          // left side of the mountain so search space is on the right
        st = mid+1;
    } else {                          // right side of mountain search space is on the left
        end = mid-1;
    }
}
return mid;
}
```