PD Algo:

Modify Binary Search:

mid = size / 2

Split in half

check mid

1) if ~~mid is~~ mid > m+1

&& mid > m-1

then mid is P

2) if mid-1 > mid > m+1

then toss everything ~~right~~ left of mid and ~~split everything~~ repeat process on ~~left~~ right of mid

3) if m+1 > mid > mid-1

then toss right of mid

and repeat on ~~right~~ left of mid

Runtime: O(log n)

input size gets halved each time O(log n)

computing mid O(1)

checking mid+1 && mid-1 O(1)

comparing O(1)

Proof: Induction

BC: $n=3$

numbers

we have 3 ~~[illegible]~~ set mid = ~~to~~ middle then look at both $m+1$ and $m-1$ p is bound to be 1 of these so we return p

IH: Assume true for all $0 \leq n \leq k$

IS: Prove ~~and~~ for $k+1$ ~~is~~

Our algorithm divides each subset in half so if we ~~[illegible]~~ have p in the left subset we know every thing to the right of mid is decreasing and we proceed to ~~search~~ ~~[illegible]~~ repeat the process on the left subset

if we have p in the right subset then everything in ~~[illegible]~~ left subset is in increasing order so we throw away left subset and repeat on right.

p is bound to be in one of these subsets. by checking mid, mid-1, and mid+1 we know ~~[illegible]~~ ~~[illegible]~~ where p lies relative to mid and can decide which half to throw away.

We can do this down to the base case where p is bound to be mid, mid+1, or mid-1