Algorithm expSearch:

Problem: find a target element t in a sorted array A if it exists.

Input: an array A, a target t, left=1 and right=n
Output: an index for t if it exists, else return -1.

Start.

if(left>right) then
  return -1
fi
m:=(left+right)÷2
if (A[m]=t) then
  return m; fi
elif (A[m]>t) then
  return expSearch(A, t, left, m-1)
file
else then
  return expSearch(A, t, m+1, right)
esle

End.

Hyp: Let P(n) be the assertion that expSearch works ∀ inputs where right-left+1=n

Base Case: when n=0, left=right=m if t=A[m] this works, if t≠A[m] then the function runs until reaching the else statement where left=m+1 ∴ left>right returning -1 if t does not exist so this works for n=0.

Inductive Step: We assume expSearch works as long as left-right+1≤k so we must prove left-right≤k there are 4 cases, where t=A[m], t>A[m], t<A[m], t does not exist in A.