Using *Introduction to the Theory of Computation, Chapter 2*
Outline

binary search
recursive binary search

\[ 0 \leq b \leq e < \text{len}(A) \]
array els & x are comparable
\[ A \text{ is sorted non-decreasing} \]
def recBinSearch(x, A, b, e):
  if b == e:
    if x <= A[b]:
      return b
    else:
      return e + 1
  else:
    m = (b + e) // 2 # midpoint
    if x <= A[m]:
      return recBinSearch(x, A, b, m)
    else:
      return recBinSearch(x, A, m+1, e)
  
  \[ \text{post-terminate} \]
  \[ \text{return } p \]
  \[ b \leq p \leq e+1 \]
  \[ b < p \Rightarrow A[p] \geq x \]
  \[ b \leq p \Rightarrow A[p] \geq x \]
conditions, pre- and post-

**pre**

- $x$ and elements of $A$ are comparable
- $e$ and $b$ are valid indices, $b \leq e$
- $A[b..e]$ is sorted non-decreasing

**post condition**

RecBinSearch($x, A, b, e$) terminates and returns index $p$

- $b \leq p \leq e + 1$
- $b < p \Rightarrow A[p - 1] < x$
- $p \leq e \Rightarrow x \leq A[p]$  

(except for boundaries, returns $p$ so that $A[p - 1] < x \leq A[p]$)
precondition $\Rightarrow$ termination and postcondition

Proof: induction on $n = e - b + 1$

$\quad b < p \Rightarrow A[p-1] < x \checkmark \quad \quad p \leq e \Rightarrow x \leq A[p]$ \checkmark

Base case, $n = 1$: Terminates because there are no loops or further calls, returns $x \leq A[b = p] \iff p = b = e$ is returned. $x > A[b = p - 1] \iff p = b + 1$ returned, so postcondition satisfied. Notice that the choice forces if-and-only-if.

Induction step: Assume $n > 1$ and that the postcondition is satisfied for inputs of size $1 \leq k < n$ that satisfy the precondition. Call RecBinSearch($A,x,b,e$) when $n = e - b + 1 > 1$. Since $b < e$ in this case, the test on line 1 fails, and line 7 executes. Exercise: $b \leq m < e$ in this case. There are two cases, according to whether $x \leq A[m]$ or $x > A[m]$.
Case 1: $x \leq A[m]$

- Show that IH applies to RBS(x,A,b,m)
- Translate the postcondition to RBS(x,A,b,m)
  - terminates, returns $p$, $b \leq p \leq m + 1$
    - $b < p \Rightarrow A[p-1] < x$
    - $p \leq m \Rightarrow A[p] \geq x$
- Show that RBS(x,A,b,e) satisfies postcondition
  - termination (from IH)
  - $m + 1 \leq e \leq e + 1 \Rightarrow b \leq p \leq m + 1 \leq e + 1$ (by IH)
  - $b < p \Rightarrow A[p] \geq x$
  - $\beta \geq p \leq e + 1$, by IH, so must show
    - $A[p] \geq x$, $p = m$, then $x \leq A[p]$ by IH
    - $p = m + 1$, then $x \leq A[m] \leq A[p]$
Case 2: \( x > A[m] \)

1. \( 1 \leq e - (m+1) + 1 < e - b + 1 \)
   \[ m - b \geq 0 \]

\[
\begin{align*}
1 \leq e - m & \quad \forall e \geq m \\
& \text{IH, IH says:}
\end{align*}
\]

- Show that IH applies to \( \text{RBS}(x,A,m+1,e) \)
- Translate postcondition to \( \text{RBS}(x,A,m+1,e) \)
  - Terminate with 
    - \( m+1 < p \Rightarrow A[p-1] < x \)
    - \( p \leq e \Rightarrow A[p] \geq x \)

- Show that \( \text{RBS}(x,A,b,e) \)
  - Termination: \( b \leq m+1 \leq p \leq e+1 \) by IH and \( m \geq b \)
    - \( p \leq e \Rightarrow A[p] \geq x \) directly from IH
    - \( b < p \Rightarrow b = m < m+1 \) must show
      - \( A[p-1] \leq x \)
      - \( m+1 < p \), then \( A[p-1] \leq x \) by IH
      - \( m+1 = p \), then \( A[p-1] = A[m] < x \) by case
what could go wrong?

- \( m = \left\lfloor \frac{e + b}{2.0} \right\rfloor \)

- \( x < A[m] \)

- ...

- Either prove correct, or find a counter-example
recursive and iterative
mergesort

MergeSort(A, b, e):
1. if b == e: return
2. m = (b + e) / 2  # integer division
3. MergeSort(A, b, m) \(1 \leq m - b + 1 < e - b + 1\)
4. MergeSort(A, m+1, e) \(1 \leq e - m < e - b + 1\)
5. for i = b,...,e: B[i] = A[i]
6. c = b
7. d = m+1
8. for i = b,...,e:
   if d > e or (c <= m and B[c] < B[d]):
     A[i] = B[c]
   else:
     A[i] = B[d]
   c = c + 1
else:
   A[i] = B[d]
9. i = i+1
10. A[i] = B[c]
11. c = c + 1
13. d = d + 1
conditions, pre- and post-

\[ p_{\text{pre}} \]

- \( b \) and \( e \) are natural numbers, \( 0 \leq b \leq e < \text{len}(A) \).
- Elements of \( A \) are comparable

\[ \checkmark \quad \text{post terminates} \]

- \( A'[b..e] \) contains the same elements as \( A[b..e] \), but sorted in non-decreasing order (use notation \( A' \) for \( A \) after calling \( \text{MergeSort}(A,b,e) \)). All other elements of \( A' \) are unchanged from \( A \).
Proof of correctness of MergeSort(A,b,e)
by induction on $n = e - b + 1$ for all arrays of size $n$,
(precondition+execution)$\Rightarrow$(termination+postcondition)

Base case, $1 = e - b + 1$: Assume MergeSort(A,b,e) is called
with $\text{len}(A) = 1$ preconditions satisfied. Then $0 \leq e \leq b \leq 0$,
so $e = b$, and the algorithm terminates with a (trivially)
sorted $A'$, satisfying the precondition.

Induction step: Assume $n \in \mathbb{N}$, $n > 1$, and for all natural
numbers $k$, $1 \leq k < n$, that MergeSort on all arrays of size $k$
that satisfy the precondition and run will terminate and satisfy
the postcondition. Assume MergSort(A,b,e) is executed and
$n = e - b + 1$. 
The test on line 1 fails, and $m$ is set to $(b + e)/2$, strictly less than $e$ (exercise).

Does the IH apply to $\text{MergeSort}(A, b, m)$ and $\text{MergeSort}(A, m+1, e)$? Translate the IH into postconditions for $\text{MergeSort}(A, b, m)$ and $\text{MergeSort}(A, m+1, e)$.

Now we need iterative correctness for the merge...
iterative correctness

partial correctness plus termination

- Preconditions plus termination imply the postcondition.
  Probably needs a loop invariant

- termination — construct a decreasing sequence in \( \mathbb{N} \).