

<div style="display: flex; justify-content: space-between; align-items: center;"> <div>—</div> <div>MODULE <i>BinarySearch</i></div> <div>—</div> </div> <div style="padding: 10px 0;"> <p>EXTENDS <i>TLC, Sequences, FiniteSets, Naturals, Integers</i></p>   <p>CONSTANTS <i>OrderedList, SearchItem</i></p> <p>VARIABLES <i>searchIndex</i></p>   <p>ASSUME <math>\wedge (Len(OrderedList) \neq 0)</math></p> <p>ASSUME <math>\wedge (SearchItem \in Nat)</math></p>   <p><math>vars \triangleq \langle searchIndex \rangle</math></p>   <p><math>InitialStart \triangleq 1</math></p> <p><math>InitialEnd \triangleq Len(OrderedList)</math></p>   <p><math>Init \triangleq</math>  <math>\quad \wedge searchIndex = 0</math></p>   <p>RECURSIVE <math>Search(-, -, -)</math></p> <p><math>Search(start, end, item) \triangleq</math>  <math>\quad \wedge LET \ Index \triangleq (start + end) \div 2</math>  <math>\quad IN \quad \vee \ \wedge start \leq end</math>  <math>\quad \quad \wedge IF \ OrderedList[Index] = item</math>  <math>\quad \quad \quad THEN \ searchIndex' = Index</math>  <math>\quad \quad \quad ELSE \ IF \ OrderedList[Index] &lt; item</math>  <math>\quad \quad \quad \quad THEN \ Search(Index + 1, end, item)</math>  <math>\quad \quad \quad \quad ELSE \ Search(start, Index - 1, item)</math>  <math>\quad \vee UNCHANGED \ vars</math></p>   <p><math>Next \triangleq Search(InitialStart, InitialEnd, SearchItem)</math></p>   <p><math>Spec \triangleq</math>  <math>\quad \wedge Init</math>  <math>\quad \wedge \Box [Next]_{vars}</math></p> </div>
---