Box Notation for Sequences

Graphical assertion about sequence $b$. It asserts that:

1. $b[0..k-1]$ is sorted (values are in ascending order)
2. all of $b[0..k-1]$ is $\leq$ all of $b[k..\text{len}(b)-1]$

**Pro Tip #1:**

index always goes above a box, never above a line
(just like house numbers go on a house not between the houses)
Q: Indices for Box Notation

Given:
• index $h$ of the first element of a segment
• index $k$ of the element that follows that segment,

Questions:
1. How many values are in segment $b[h .. k - 1]$?
2. How many values are in $b[h .. h - 1]$?
3. How many values are in $b[h .. h + 1]$?

Pro Tip #2:
Size is “Follower minus First”
Follower: next thing outside the specified range
count num adjacent equal pairs

Approach #1: compare s[k] to the character in front of it (s[k-1])

# set n_pair to # adjacent equal pairs in s

n_pair = 0
k = 1

while k < len(s):
    if s[k-1] == s[k]:
        n_pair += 1
    k = k + 1
count num adjacent equal pairs

Approach #1: compare s[k] to the character in front of it (s[k-1])

# set n_pair to # adjacent equal pairs in s

pre: seq s \( ? \) (unknown values) \( n \geq 0, \ n\_pair = 0 \)
n_pair = 0

k = 1

INV: seq s \text{processed} \( ? \) (unknown) \( n \_pair = \text{num adjacent pairs in } s[0..k-1] \)

while k < len(s):
    if s[k-1] == s[k]:
        n_pair += 1
    k = k + 1

post: seq s \text{processed} \( n \_pair = \text{num adjacent pairs in } s[0..n-1] \)
find the max of a seq

Task: find the maximum of a sequence \( s \)

\[
k = 1 \\
big = s[0]
\]

\[
\text{while } k < \text{len}(s):
\]
\[
big = \max(big, s[k]) \\
k = k + 1
\]
Task: find the maximum of a sequence s

pre: s  
\[0, \ldots, n\]  
\text{big is the max of this segment (s[0])}

\text{? (unknown values)}  
n > 0, \text{big} = s[0]

inv: s  
\[0, k, \ldots, n\]  
\text{big is max of this segment}  
\text{?}  
n > 0, \text{big} = s[0..k-1]

while \( k < \text{len}(s) \):

\text{big} = \max(\text{big}, s[k])

\text{?}  
k = k + 1

post: s  
\[0, \ldots, n\]  
\text{big is the max of this segment}  
k = n, \text{big} = \max(s[0..n-1])
Developing Algorithms on Sequences

- Specify the algorithm by giving its **precondition** and **postcondition** as pictures.
- Draw the **invariant** by drawing another picture that “moves from” the **precondition** to the **postcondition**
  - The invariant is true at the beginning and at the end
- The four loop design questions
  1. How does loop start (how to make the invariant true)?
  2. How does it stop (is the postcondition true)?
  3. How does the body make progress toward termination?
  4. How does the body keep the invariant true?
Invariants: separate $+$ from $-$ in a list

Task: Put negative values before nonnegative ones and return the split index

5 \ -7 \ 2 \ 2 \ -1 \ 8 \ -3 \ 9 \ 3 \quad \Rightarrow \quad -7 \ -1 \ -3 \ 2 \ 5 \ 8 \ 2 \ 9 \ 3 \quad k = 3
Task: Put negative values before nonnegative ones and return the split index

Invariants: separate + from – in a list

pre: \( s \)

\( n \geq 1 \)

\( k = 0 \)

\( j = n \)

inv: \( s \)

\( < 0 \)

\( ? \)

\( \geq 0 \)

while \( k < j \):

\(<\text{body goes here}>\)

post: \( s \)

\( k = j \)
Body: separate + from – in a list

\[ k = 0 \]
\[ j = n \]

inv: \( s \)

<table>
<thead>
<tr>
<th>0</th>
<th>( k \rightarrow )</th>
<th>?</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0</td>
<td>( \geq 0 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

while \( k < j \):

if \( s[k] < 0 \): # kth elem stays where it is
\[ k = k + 1 \]
elif \( s[j-1] \geq 0 \): # (j-1)th elem stays where it is
\[ j = j - 1 \]
else: # both elements in the wrong place
swap(s, k, j-1)
\[ k = k + 1 \]
\[ j = j - 1 \]

post: \( s \)

<table>
<thead>
<tr>
<th>0</th>
<th>( k \rightarrow )</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0</td>
<td>( \geq 0 )</td>
<td>( k = j )</td>
</tr>
</tbody>
</table>