Announcements

Thursday Extras: CS Commons on Thursdays @ 4:00 pm
This week: CS Student Research Group

Grinnell ACM Chapter meeting, Thursday, 7:00 pm, Sci. 3821
Discussion of student mentoring program in local schools
Opportunities for involvement of computing education

Reminder: be sure to be prepared with readings/past work for each class day!

External Reviewers on campus: Thursday/Friday

CSC 207 tutors available
must be attending class regularly
must be coming to office hours and/or mentor sessions
talk to Sarah Dahlby-Albright and/or me for details

CSC 207.02 Schedule

loop invariants
discussion
clicker questions
Consider the problem of computing the sum of values in an int array a

3 looping constructs are proposed

1. `for (i = 0; i < a.length; i++)`
2. `for (i = 1; i < a.length; i++)`
3. `for (i = 1; i <= a.length; i++)`

Which of these could be used in a correct program?
A. 1 only
B. 2 only
C. 3 only
D. exactly two of these
E. all three
Consider the problem of computing the sum of values in an `int` array `a`.

Suppose for iteration `i` of a loop, we assume:

\[ \text{sum} = a[0] + \ldots + a[i] \]

Two loop solutions are proposed:

1. \[
\text{sum} = a[0]; \\
   \text{i} = 0; \\
   \text{while (i < a.length-1)} \\
   \{ \text{i}++; \\
       \text{sum} += a[i]; \\
   \} \\
\]
2. \[
\text{sum} = 0; \\
   \text{i} = 0; \\
   \text{while (i < a.length)} \\
   \{ \text{sum} += a[i]; \\
       \text{i}++; \\
   \} \\
\]

Which of these loops is consistent with the given loop invariant?

A. neither
B. 1 only
C. 2 only
D. both 1 and 2
Consider the problem of computing the sum of values in an int array `a`.

Suppose for iteration `i` of a loop, we assume:

`sum = a[0] + ... + a[i-1]`

Two loop solutions are proposed:

1. `sum = a[0];
   i = 1;
   while (i < a.length)
   { i++;
     sum += a[i];
   }

2. `sum = 0;
   i = 0;
   while (i < a.length)
   {  sum += a[i];
     i++;
   }

Which of these loops is consistent with the given loop invariant?

A. neither
B. 1 only
C. 2 only
D. both 1 and 2
Consider a binary search based on the following loop invariant:

\[
\begin{array}{cccc}
\text{array } & \text{lo} & \text{hi} & \text{unexamined} \\
\text{a} & < \text{item} & \text{unexamined} & > \text{item} \\
\end{array}
\]

What is the appropriate initialization of lo and hi?

A. \( \text{lo} = 0; \text{hi} = \text{a.length-1}; \)
B. \( \text{lo} = 0; \text{hi} = \text{a.length}; \)
C. \( \text{lo} = 1; \text{hi} = \text{a.length-1}; \)
D. \( \text{lo} = 1; \text{hi} = \text{a.length}; \)

What is the proper condition for the overall loop?

A. \( \text{while} \ (\text{lo} < \text{hi} - 2) \)
B. \( \text{while} \ (\text{lo} < \text{hi} - 1) \)
C. \( \text{while} \ (\text{lo} < \text{hi}) \)
D. \( \text{while} \ (\text{lo} < \text{hi} + 1) \)
Consider a binary search based on the following loop invariant:

<table>
<thead>
<tr>
<th>array a</th>
<th>&lt; item</th>
<th>unexamined</th>
<th>&gt; item</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo</td>
<td></td>
<td></td>
<td>hi</td>
</tr>
</tbody>
</table>

What is the appropriate initialization of lo and hi?

A. `lo = 0; hi = a.length - 1;`
B. `lo = 0; hi = a.length;`
C. `lo = 1; hi = a.length - 1;`
D. `lo = 1; hi = a.length;`

What is the proper condition for the overall loop?

A. `while (lo < hi - 2)`
B. `while (lo < hi - 1)`
C. `while (lo < hi)`
D. `while (lo < hi+1)`