CSC207.01 2013F, Class 38: Detour: Anonymous Inner Classes

Overview

- Preliminaries.
    - Admin.
    - Questions on HW9.
- Detour: An invariant for `merge`.
- Anonymous inner classes.
- Lab.

Preliminaries

Admin

- I may not have a reading for tomorrow. (Given that I’m mostly booked until 8 pm, it’s unlikely.)
- Upcoming extra credit opportunities:
    - G-Tones, Train Opponents, and Noteworthy, 9pm in Bucksbaum Rotunda
    - Town Hall, Wednesday, November 13, noon or 7:30 p.m.
    - Learning from Alumni, Thursday, 2:15: Atul Gupta ’88, Trustee
    - CS Extra, Thursday, 4:30: Hilary Mason ’00
    - Informal chat with Hilary Mason, Thursday, 8-9 pm, Commons
    - CS Table, Friday, HCI
    - Innovation Class, Friday, 12:45, ARH 302, Hilary Mason ’00
    - CLS with Hilary Mason ’00, Friday, 4:15-6:30 (I don’t know how career connections work)
    - Swim meet, Saturday, noon
    - Digital Commons talk Monday, November 19, 7:00 p.m. or so
    - "Data Sovereignty: The Challenge of Geolocating Data in the Cloud", November 25, 4:15 JRC 101
    - "Gold Fever" by Andrew Sherburne ’01 or so, 7:00 p.m., Monday, November 25, ARH 302
    - Tuesday, November 20, 4:15 p.m., JRC 209 a gaming event with the game [d0x3d!]

Questions on HW9

- Do you want the analysis for HW9?
    - It would be nice, but it’s not necessary.
- For the ones that divide by 2, powers of two are good
- Should we use BigDecimal for the calculator?
    - Nah. Just double is fine.

Detour: An invariant for `merge`

Remind me to end by 10:25

Background:

- Preconditions:
  - The subarray of a1 from lb1 (inclusive) to ub1 (exclusive) is sorted
  - The subarray of a2 from lb2 (inclusive) to ub2 (exclusive) is sorted
- Postconditions:
  - target is a permutation of the combination of the two subarrays
  - target is sorted
- We will rely on the preconditions to make sure our invariant holds at the beginning of the loop.
- We will try to achieve the postconditions at the end of our loop.

A picture of the state of our system

```
a1:
+------------+-----------+-------------+------------+
| irrelevant | processed | unprocessed | irrelevant |
+------------+-----------+-------------+------------+
0            lb1         c1            ub1          a1.length

a2:
+------------+-----------+-------------+------------+
| irrelevant | processed | unprocessed | irrelevant |
+------------+-----------+-------------+------------+
0            lb2         c2            ub2          a1.length

target:
+-------------------------+------------------------+
|         processed       |         "empty"        |
+-------------------------+------------------------+
0                         i                        target.length
```

Our focus should probably be the processed values in target. What can we say about that section? What should we say? (We'll work both formally and informally.)

- a1 is sorted from lb1 to ub1
- a2 is sorted from lb2 to ub2
- The "processed" section of target is sorted
  - For all j, 0 < j < i, order.compare(target[j-1], target[j]) <= 0
- The "processed" section of target is a permutation of the processed sections of a1 and a2.
- All of the "processed" elements of target are less than or equal to all of the "unprocessed" elements of a1 and a2.

Is that enough? What's the basic step we expect to do?

Action: Copy the smaller of a1[c1] and a2[c2] to target[i]

```
if (order.compare(a1[c1], a2[c2]) <= 0) {
    target[i++] = a1[c1++];
} else {
    target[i++] = a2[c2++];
} // else: a2[c2] is smaller
```

When do we stop? When we run out of elements in a1 or a2

```
a1:
+------------+-----------+-------------+
| irrelevant | processed |  irrelevant |
+------------+-----------+-------------+
0            lb1         c1,ub1        a1.length

a2:
+------------+-----------+-------------+------------+
| irrelevant | processed | unprocessed | irrelevant |
+------------+-----------+-------------+------------+
0            lb2         c2            ub2          a1.length

target:
+---------------------------+-------------------------+
|         processed         |         "empty"         |
+---------------------------+-------------------------+
0                           i                         target.length
```

Whoops! Haven’t quite achieved the postcondition.

- But, we know that everything in target.processed is sorted, everything in a2.unprocessed is sorted, and everything in target.processed is smaller than a2.unprocessed, so COPY THEM OVER!

Anonymous inner classes

Lab