CSC207.01 2014S, Class 21: Exam 1 Discussion

Overview

- Preliminaries.
  - Admin.
  - Upcoming work.
  - Questions.
- Problem 1 - Testing.
- Problem 2 - Averaging.
- Problem 3 - Stringing.
- Problem 4 - Replacing.
- Additional issues.
- Makeup.

Preliminaries

Admin

- Exam 1 will be returned at the end of class.
- MathLAN may or may not be acting up again. But we have no lab today, so it's okay.
- Our graders are working hard, but they've discovered just how much time it takes to grade well.
- Extra credit:
  - Town hall meeting Tuesday at noon in JRC 101 or 7:30pm in Harris.
  - Wellness fair.
  - CS Extras: The new CS Curriculum.
  - More?

Upcoming work.

- Reading for Tuesday: Anonymous inner classes.
- Homework 5 is due *next* Wednesday (March 5), but you may need two weeks to get it done.
- Makeup for exam 1 is due Sunday night.

Questions on the homework?

Problem 1 - Testing

Normal cases

- A bunch of identical positive numbers
- A bunch of different positive numbers
- Mixed positive and negative numbers
  - Negative result
  - Positive result
  - Zero result
- A bunch of identical negative numbers
- A bunch of different negative numbers
- Small arrays and big arrays
  - Really big arrays

Particular tests that we found useful

- [1,-2,3,-4,5,-6]
- [1,2,3,4,5]
- [-2,1]

Special cases

- Rounding
  - test(1/2, average([0,1]))
  - test(3/2, average([1,2]))
  - Down for positive, up for negative
  - Using a variety of fractional portions

Edge cases

- Mixing things near Long.MAX*VALUE and Long.MIN*VALUE
- *Near* Long.MAX*VALUE and *Near* Long.MIN*VALUE
- A really really really big array (say more than Integer.MAX*VALUE elements, all of them near Long.MAX*VALUE)
- Does it throw exceptions
  - (Beyond scope of what I expected.)

Problem 2 - Averaging

- Using doubles.

Problem 3: Stringify

- It might have been helpful to write a helper that does the conversion in memory, before writing to the file.

- What do you see as the difference between

```
String line = infile.readLine();
while (line != null)
  {
    ...
    line = infile.readLine();
  } // while
```

and

```
String line;
while ((line = infile.readLine()) != null)
  {
    ...
  } // while
```

- What do you see as the difference between

```
ch = line.charAt(i);
if (ch == '\t')
  ...
else if (ch == '\n')
  ...
else if (ch == '\\')
  ...
else if (ch == '\"')
  ...
else
  ...
```

and

```
switch (ch = line.charAt(i))
  {
    case '\t':
      ...;
      break;
    case '\n':
      ...;
      break;
    case '\\':
      ...;
      break;
    case '\"';
      ...;
      break;
```

```
      default:
        ...;
        break;
    } // switch
```

Why did you write `"\\" + "n"` rather than `"\\n"`?

- Computers are evil and failed to work appropriately for the second.

Problem 4 - Replacing

- Some of you used `String.replace`. I thought the problem was pretty clear that that approach was not permitted.
- Everyone who did a real solution should have written a helper method that checked whether the string matched at the given position.

- Most of you used (approximately)

```
int i = 0;
while (i < contents.size-pattern.length())
  {
    if (this.matchesAt(i, pattern))
      {
        this.remove(i, i + pattern.length());
        this.prepend(i, replacement);
        i += pattern.length();
      } // if match at position i
    else // if no match at position if
      {
        i++;
      } // if no match at position i
  } // while
```

- This solution is potentially quadratic in the size of the input (e.g., `"aaaaaa".replace("a","b")`)

Other Issues

- Formatting
- Tarballs
- Sam’s errors

Makeup

Although I said that I was not going to allow a makeup exam, there were enough problems on problems 2 and 4 that I will allow you to make those problems up for partial credit.