As many of you noticed, the bytecodes that were given to you for Assignment #5 were not sufficient to write a binary search program.

You need 2 more bytecodes, namely IALOAD and IASTORE to load/store to/from the stack in a location in local storage whose address is computed rather than a constant.

More precisely;

IALOAD (bytecode 0x2e) takes the top of the stack as an index in the local storage area and loads the contents of that local storage word on top of the stack (destroying the index value).

That is if the local storage area looks like (the left column is the index, the right column is the contents of the local storage area):

```
            ----------
  0             17
            ----------
  1             73
            ----------
  2              4
            ----------
  3             27
            ----------
  4             13
            ----------
 etc.
```

and if the stack looks like

```
            ----------
                19        <---- "below" top of the stack
            ----------
                 3        <---- top of the stack
            ----------
```

The bytecode IALOAD will result in the new stack configuration

```
            ----------
                19
            ----------
                27       <---- top of the stack
            ----------
```

IASTORE (bytecode 0x4f) stores the element below the top of the stack at the local storage location whose index is on the top of the stack. Then the two top entries on the stack are popped.

That is with the initial conditions:

```
           ----------
  0            17
           ----------
  1            73
           ----------
  2             4
           ----------
  3            27
           ----------
  4            13
           ----------
 etc.
```

and

```
           ----------
              214
           ----------
               19         <---- "below" top of the stack
           ----------
                3         <---- top of the stack
           ----------
```

IASTORE would result in

```
           ----------
  0            17
           ----------
  1            73
           ----------
  2             4
           ----------
  3            19
           ----------
  4            13
           ----------
 etc.
```

and

```
           ----------
              214        <---- top of the stack
           ----------
```

Thanks to all of you who noticed this problem.

Note that you can create your own local variables. If "key" is at location 0 (in local storage), "size" at location 1, and "array" at the next "size" locations, the locations "size + 2", "size + 3" etc can be used to store temporary variables, if need be.

In the "real" JVM IALOAD and IASTORE are slightly more complex because the local storage area is more complex. You have to identify the "object" (e.g., array) that you want to index.

JLB and Saurabh