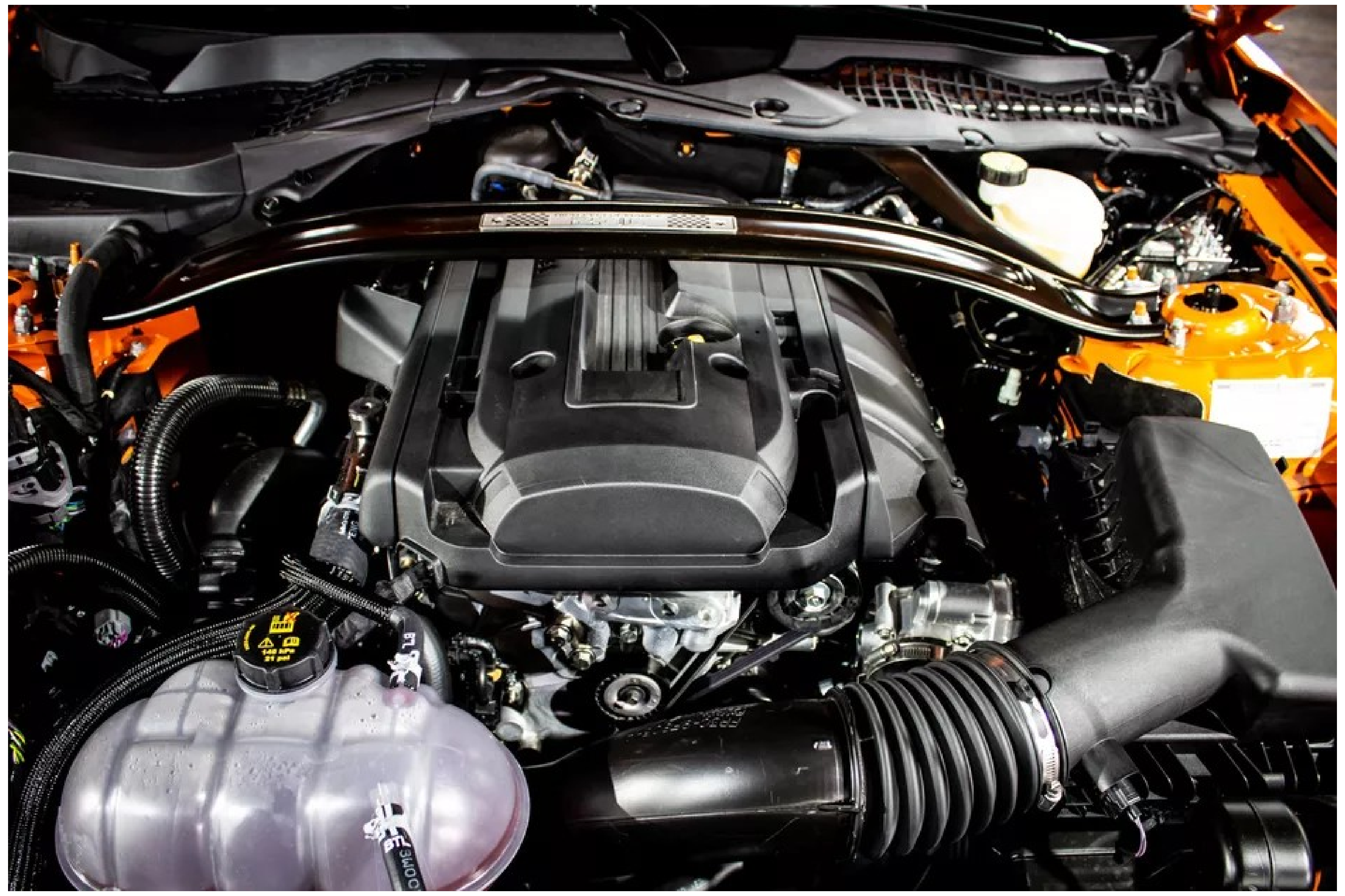
Just-In-Time Compilation - 1

With Cliff Click

Class Goals



Learn basic concepts

- Compilation is a full 4-year education program!
- Learn how the basic JIT'ing process works
- Learn what & why JIT optimizations
- Look at what the JIT will NOT do

Limits of class time:

- Not: "how to tune the JIT" – but yes, understand the basic terms
- Not: "how to write fast code" – but yes, understand one of major players in fast code

JIT Compilation

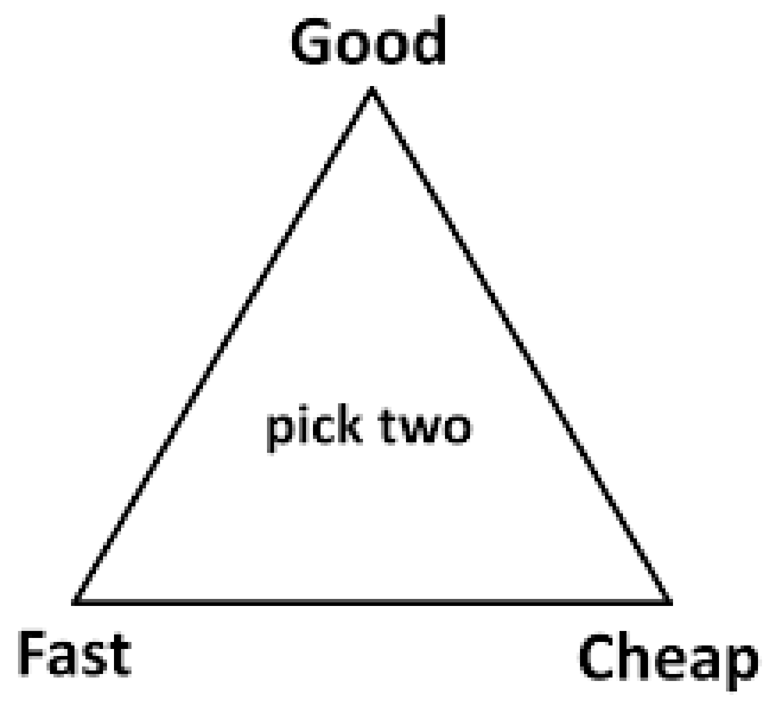
It’s just like Compilation! While You Wait!

- Tradeoffs in code-speed vs time-to-code-ready

Typically several JIT stages

- Interpret / template-style JIT, soonest ready for 10% speed
- “Stage 1” / mid-tier, ready in ~1ms/bytecode for 70% speed
- “Stage 2” / heavy-tier, ready in 10ms/bytecode for 100% speed

Both JVM and Java Script run tiered JIT’d compilation



First stage of a rocket detaches and falls away as second stage fires and moves ahead.

[redacted]

Interpreter / Template JIT

Good
pick two
Fast
Cheap

“See, Do” execution

- See bytecode, Do bytecode. About ½ the cost is “See” and ½ is “Do”.

  Least infrastructure, easiest to implement

- Machine code “Do” snippets per-bytecode, and more to “See”. Basic profiling.

Template Style: “See” all first, then “Do” all

- “See” all bytecodes, copying code-snippets next to each other
- “Do” no longer has to “See”, so about 2x faster **but** slower setup

Interpreter beats Template for run-once code

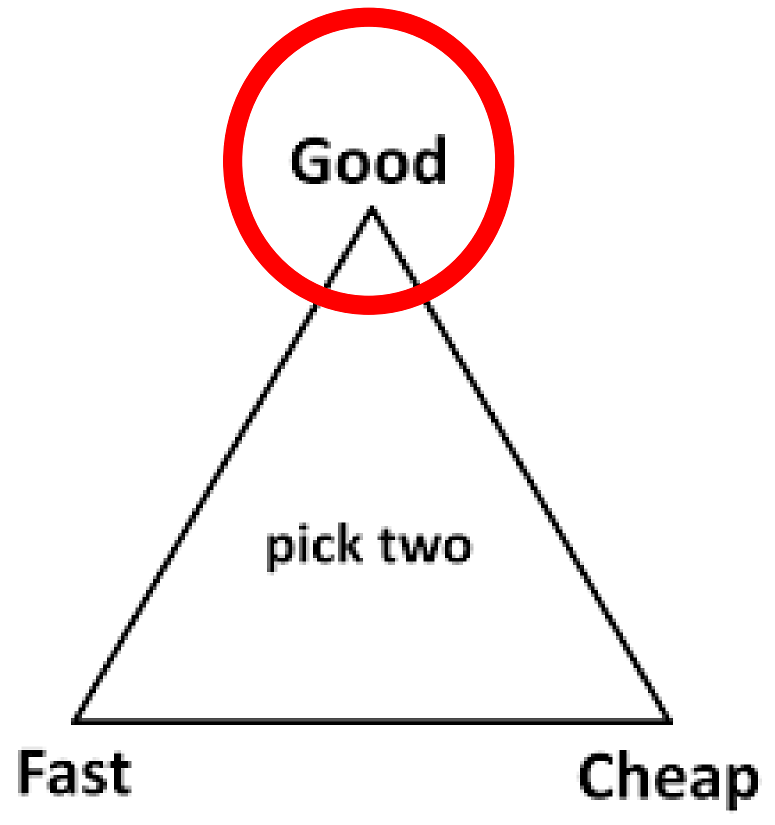
Template beats Interpreter for run ~3+ times

Stage-1 Compilation

Convert bytecodes to Intermediate Representation

- Way easier to manipulate!
- Basic/fast optimizations. Limited “heroic” opts. Small inlining.
- Convert again to machine code. Fast/dumb register allocation.
- More complex lifetime management. Smart profiling.
- Much slower setup vs Interpreter/Template

Nice speedup over Interpreter/Template



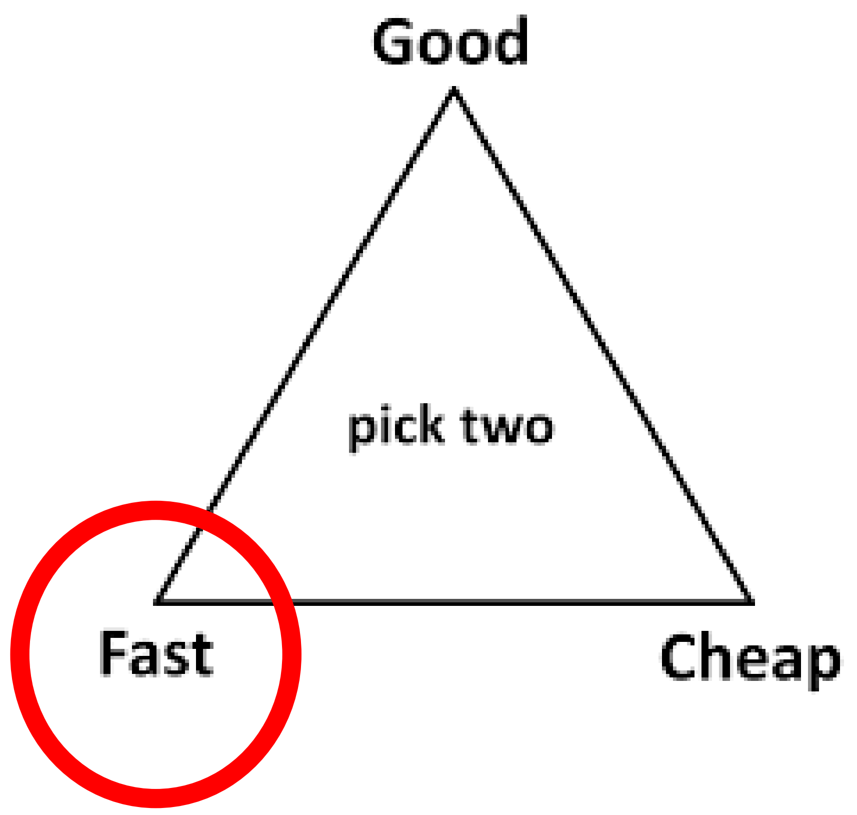
[redacted]

Stage-2 Compilation

ALL the bells and whistles (given time budget)

- All optimizations, all “heroic” opts. Massive inlining.

  Profile-based optimizations & much more.
- Convert again to machine code. Slow/smart register allocation.
- Much more complex lifetime management: heroic opts can fail.
- Much slower Stage-1, and again a nice speedup



Final speed very competative with native C

Running Example

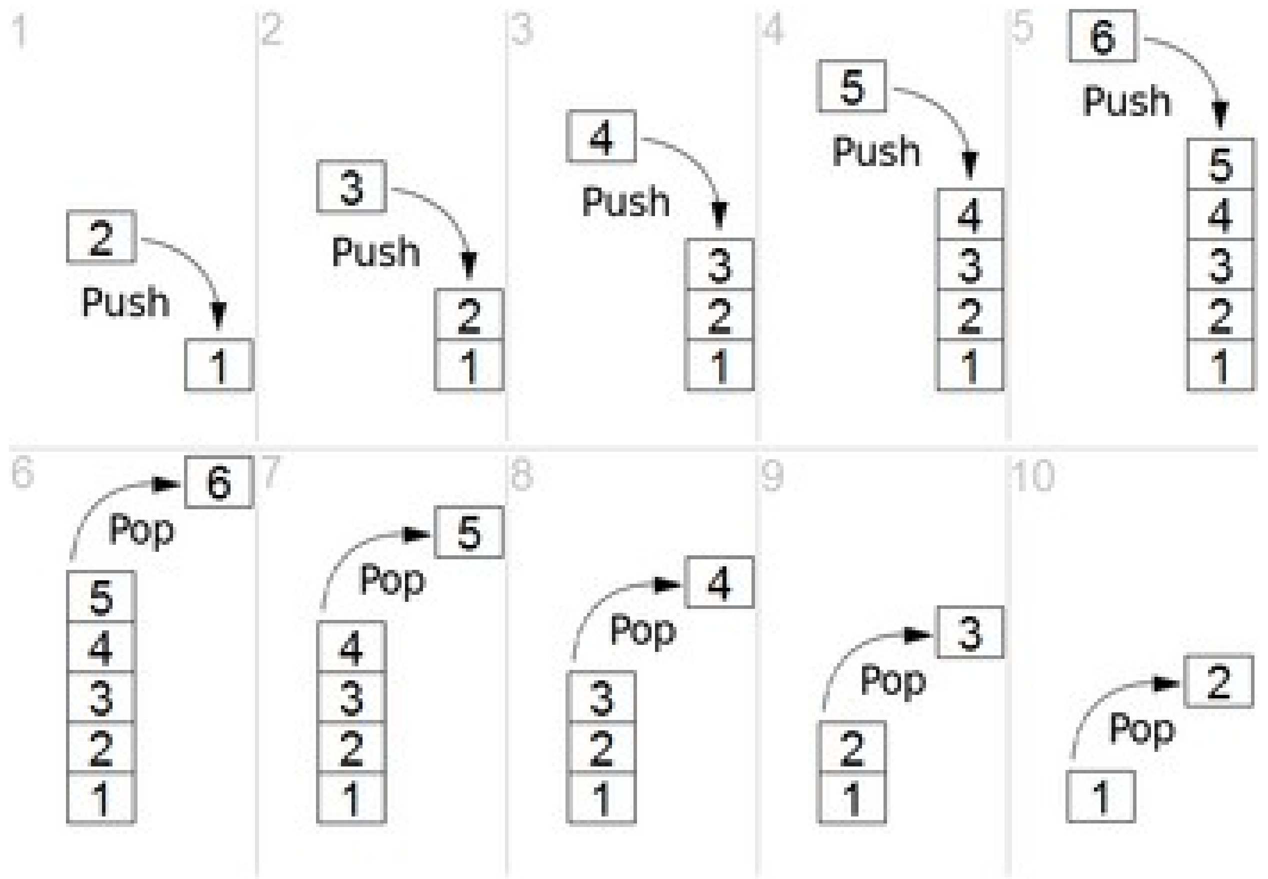
String.hashCode

- Modified slightly to be a better example

Interpeter:

- “Stack” model + locals
- Bytecodes push/pop
- Some local “registers”

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```



Bytecodes

Execution Model:

- Stack machine + registers
- Bytecodes as “assembly”

Interpreter:

- Method+Code pointer
- “See” bytecode: fetch & dispatch
- “Do” bytecode
- Repeat forever

```
 0: aload_0        // Push 'this'
 1: getfield  #2   // Push 'hash'
 4: istore_1       // reg1 = h = hash
 5: iload_1        // Push 'h'
 6: ifne      43   // early exit if hash!=0
 9: iconst_0       // 0
10: istore_2       // reg2 = i = 0
// LOOP
11: iload_2        // Push i
12: aload_0        // Push 'this'
13: getfield  #3   // 'value'
16: arraylength    // value.length
17: if_icmpge 38   // Normal exit if i>=value.length
20: bipush    31   // 31
22: iload_1        // h
23: imul           // h*31
24: aload_0        // Push 'this'
25: getfield  #3   // 'value'
28: iload_2        // Push 'i'
29: caload         // value[i]
30: iadd           // h*31 + value[i]
31: istore_1       // h = h*31 + value[i]
32: iinc 2, 1      // reg2+=1; i++
35: goto      11   // Loop repeats

38: aload_0        // Push 'this'
39: iload_1        // Push 'h'
40: putfield  #2   // hash = h;

43: iload_1        // Push 'h'
44: ireturn        // return h;
```

[redacted]

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : ?
2: i   : ?
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield  #2     // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne      43     // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield  #3     // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush    31     // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield  #3     // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto      11     // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield  #2     // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : ?
2: i    : ?
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0            // Push 'this'
 1: getfield   #2      // Push 'hash'
 4: istore_1           // reg1 = h = hash
 5: iload_1            // Push 'h'
 6: ifne       43      // early exit if hash!=0
 9: iconst_0           // 0
10: istore_2           // reg2 = i = 0
// LOOP
11: iload_2            // Push i
12: aload_0            // Push 'this'
13: getfield   #3      // 'value'
16: arraylength        // value.length
17: if_icmpge 38       // Normal exit if i>=value.length
20: bipush     31      // 31
22: iload_1            // h
23: imul               // h*31
24: aload_0            // Push 'this'
25: getfield   #3      // 'value'
28: iload_2            // Push 'i'
29: caload             // value[i]
30: iadd               // h*31 + value[i]
31: istore_1           // h = h*31 + value[i]
32: iinc 2, 1          // reg2+=1; i++
35: goto       11      // Loop repeats

38: aload_0            // Push 'this'
39: iload_1            // Push 'h'
40: putfield   #2      // hash = h;

43: iload_1            // Push 'h'
44: ireturn            // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : ?
2: i    : ?
  Stack
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    : ?
2: i    : ?
  Stack
   0
```

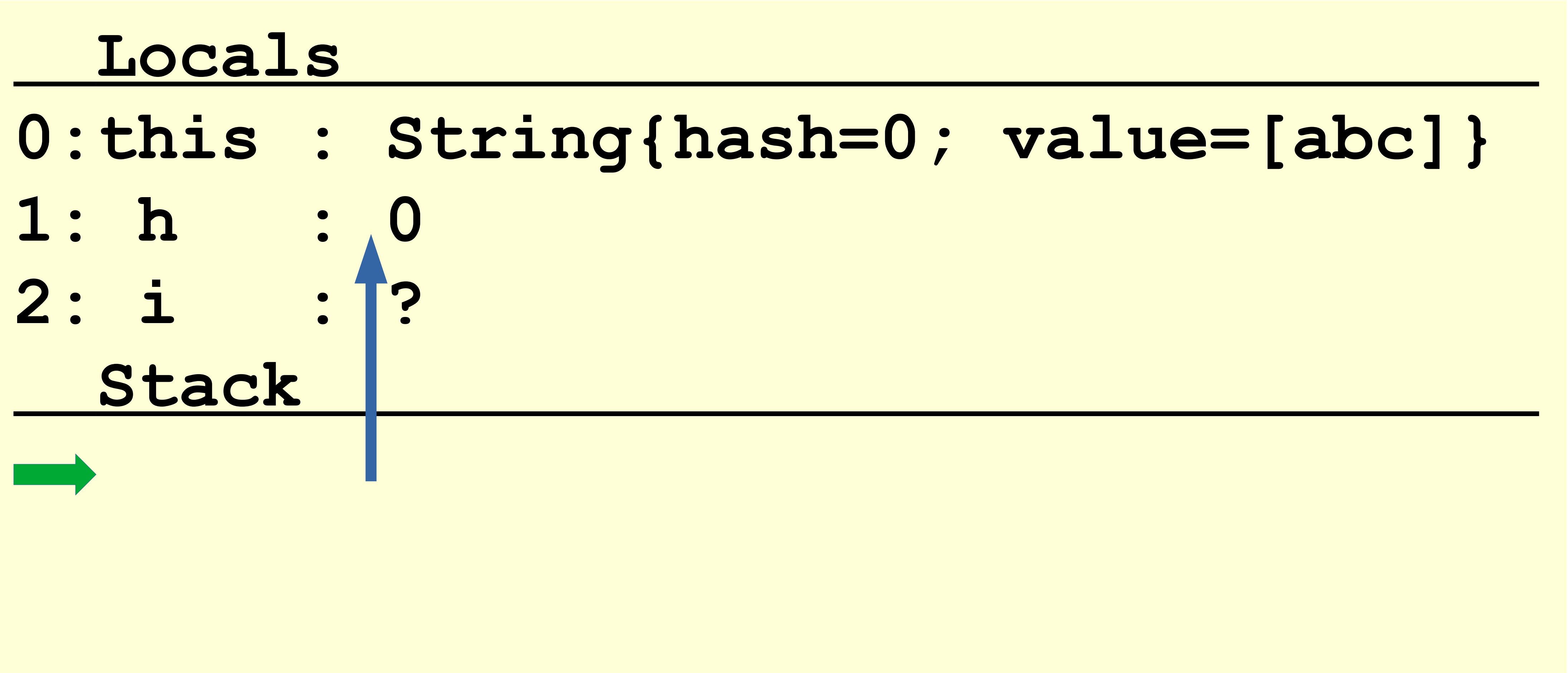
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield    #2    // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne        43    // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield    #3    // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush      31    // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield    #3    // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto        11    // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield    #2    // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter



```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : ?
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : ?
  Stack
   0
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : ?
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield    #2   // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne        43   // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield    #3   // 'value'
16: arraylength       // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush      31   // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield    #3   // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto        11   // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield    #2   // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  ?
  Stack
   0
```

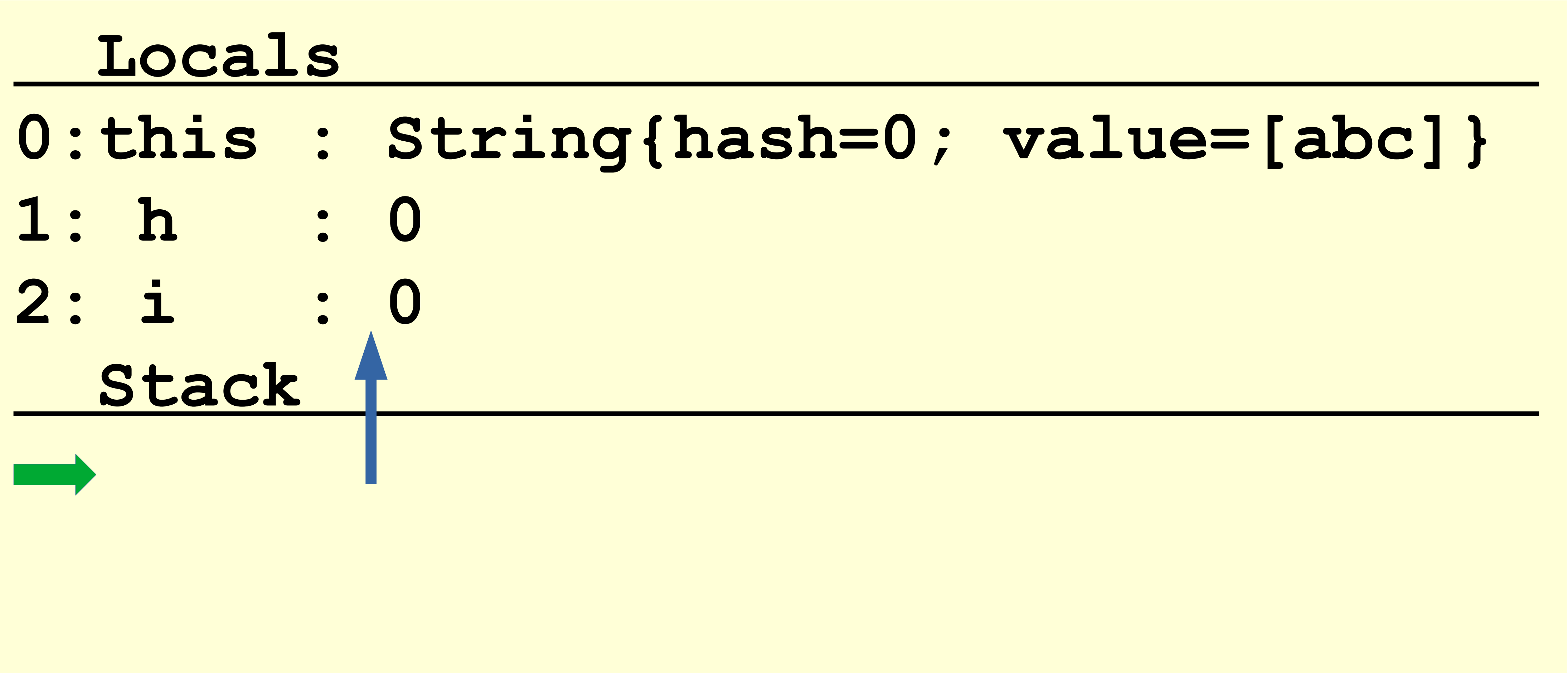
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield    #2   // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne        43   // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield    #3   // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush      31   // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield    #3   // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto        11   // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield    #2   // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter



```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : 0
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // local#1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // local#2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : 0
  Stack
   0
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield    #2   // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne        43   // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield    #3   // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush      31   // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield    #3   // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto        11   // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield    #2   // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : 0
2: i   : 0
  Stack
   0
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  0
  Stack
    0
    [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield    #2    // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne        43    // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield    #3    // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush      31    // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield    #3    // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto        11    // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield    #2    // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : 0
  Stack
   0
   3
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 0
2: i    : 0
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0         // Push 'this'
 1: getfield   #2   // Push 'hash'
 4: istore_1        // reg1 = h = hash
 5: iload_1         // Push 'h'
 6: ifne       43   // early exit if hash!=0
 9: iconst_0        // 0
10: istore_2        // reg2 = i = 0
// LOOP
11: iload_2         // Push i
12: aload_0         // Push 'this'
13: getfield   #3   // 'value'
16: arraylength     // value.length
17: if_icmpge 38    // Normal exit if i>=value.length
20: bipush     31   // 31
22: iload_1         // h
23: imul            // h*31
24: aload_0         // Push 'this'
25: getfield   #3   // 'value'
28: iload_2         // Push 'i'
29: caload          // value[i]
30: iadd            // h*31 + value[i]
31: istore_1        // h = h*31 + value[i]
32: iinc 2, 1       // reg2+=1; i++
35: goto       11   // Loop repeats

38: aload_0         // Push 'this'
39: iload_1         // Push 'h'
40: putfield   #2   // hash = h;

43: iload_1         // Push 'h'
44: ireturn         // return h;
```

[redacted]

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  0
  Stack
   31
```

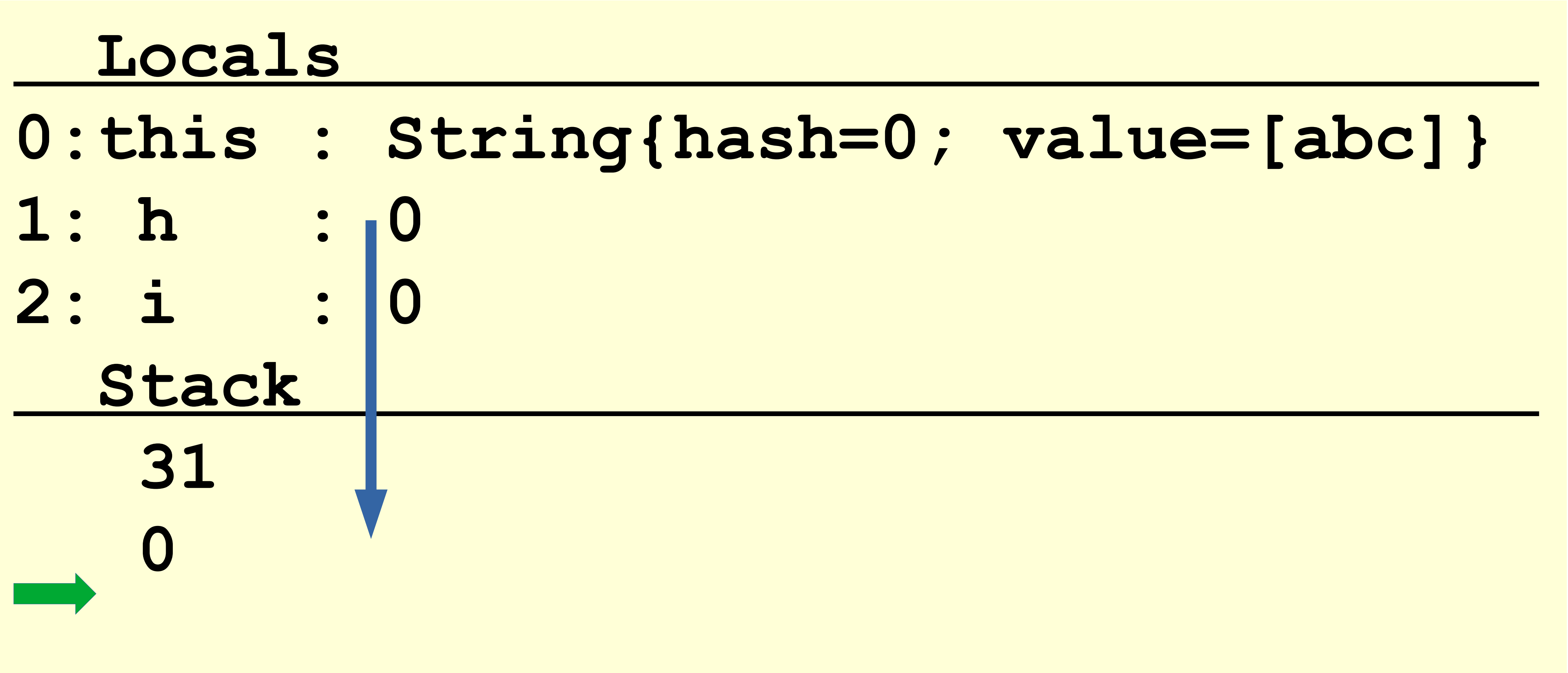
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter



```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    : 0
2: i    : 0
  Stack
   31
   0
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  0
  Stack
   0
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne        43   // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : 0
2: i   : 0
  Stack
   0
   String{hash=0; value=[abc]}
```

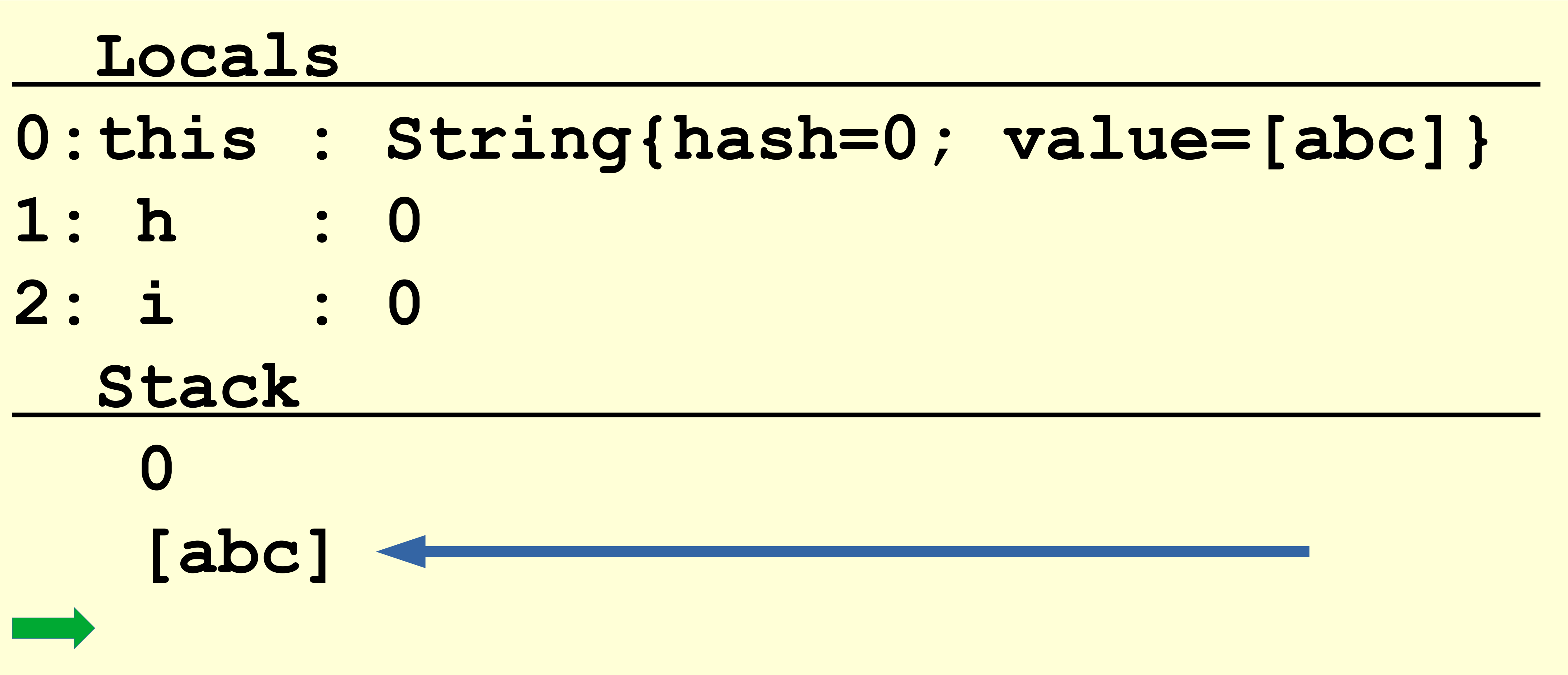
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter



```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  0
  Stack
   0
   [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  0
2: i    :  0
  Stack
   0
   [abc]
   0
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge  38    // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : 0
2: i   : 0
  Stack
   0
   97 ‘a’
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : 0
2: i   : 0
  Stack
   97
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 0
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0         // Push 'this'
 1: getfield   #2   // Push 'hash'
 4: istore_1        // reg1 = h = hash
 5: iload_1         // Push 'h'
 6: ifne       43   // early exit if hash!=0
 9: iconst_0        // 0
10: istore_2        // reg2 = i = 0
// LOOP
11: iload_2         // Push i
12: aload_0         // Push 'this'
13: getfield   #3   // 'value'
16: arraylength     // value.length
17: if_icmpge 38    // Normal exit if i>=value.length
20: bipush     31   // 31
22: iload_1         // h
23: imul            // h*31
24: aload_0         // Push 'this'
25: getfield   #3   // 'value'
28: iload_2         // Push 'i'
29: caload          // value[i]
30: iadd            // h*31 + value[i]
31: istore_1        // h = h*31 + value[i]
32: iinc 2, 1       // reg2+=1; i++
35: goto       11   // Loop repeats

38: aload_0         // Push 'this'
39: iload_1         // Push 'h'
40: putfield   #2   // hash = h;

43: iload_1         // Push 'h'
44: ireturn         // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
➡
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   1
```

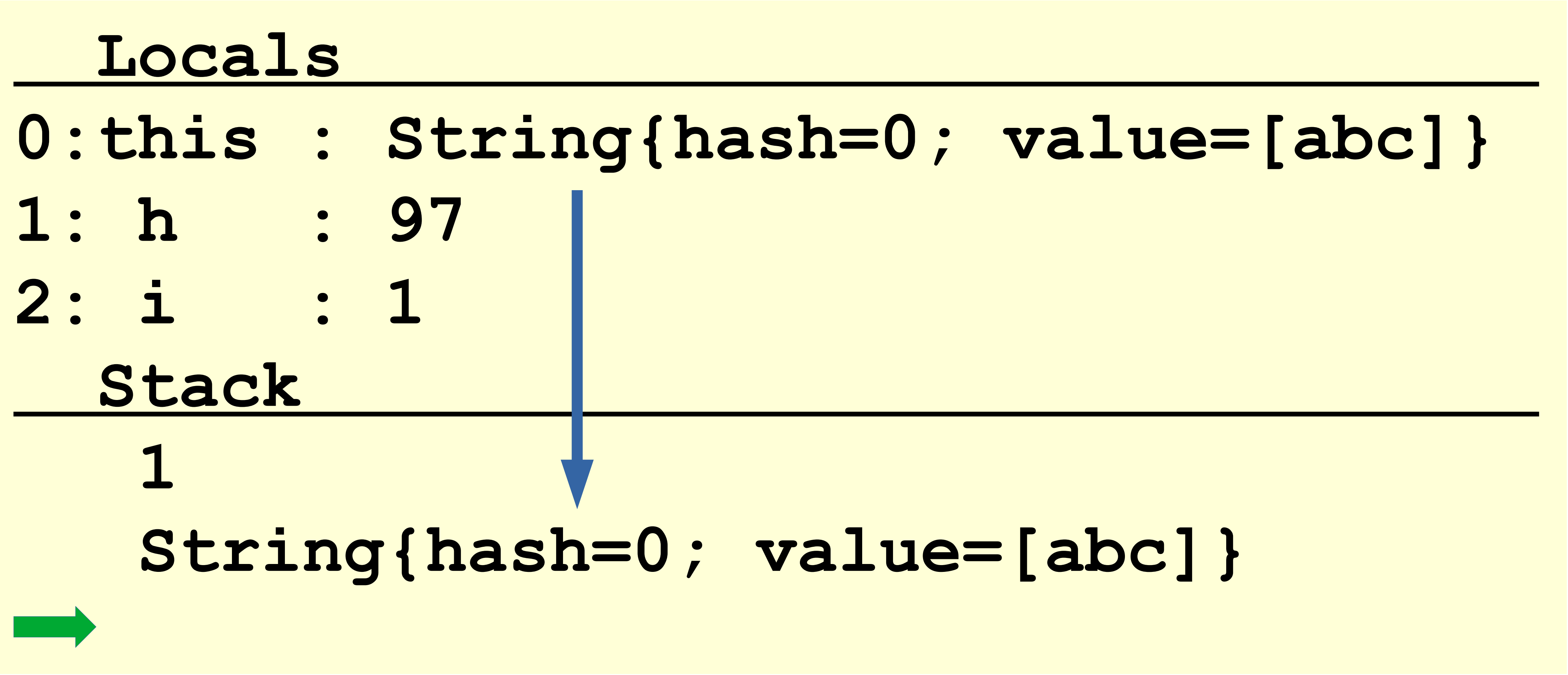
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter



```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   1
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
    1
    [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   1
   3
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield    #2    // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne        43    // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield    #3    // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush      31    // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield    #3    // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto        11    // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield    #2    // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
   31
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

[redacted]

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   31
   97
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
   3007
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0         // Push 'this'
 1: getfield   #2   // Push 'hash'
 4: istore_1        // reg1 = h = hash
 5: iload_1         // Push 'h'
 6: ifne       43   // early exit if hash!=0
 9: iconst_0        // 0
10: istore_2        // reg2 = i = 0
// LOOP
11: iload_2         // Push i
12: aload_0         // Push 'this'
13: getfield   #3   // 'value'
16: arraylength     // value.length
17: if_icmpge 38    // Normal exit if i>=value.length
20: bipush     31   // 31
22: iload_1         // h
23: imul            // h*31
24: aload_0         // Push 'this'
25: getfield   #3   // 'value'
28: iload_2         // Push 'i'
29: caload          // value[i]
30: iadd            // h*31 + value[i]
31: istore_1        // h = h*31 + value[i]
32: iinc 2, 1       // reg2+=1; i++
35: goto       11   // Loop repeats

38: aload_0         // Push 'this'
39: iload_1         // Push 'h'
40: putfield   #2   // hash = h;

43: iload_1         // Push 'h'
44: ireturn         // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   3007
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge  38     // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

[redacted]

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
   3007
   [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  97
2: i    :  1
  Stack
   3007
   [abc]
   1
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this : String{hash=0; value=[abc]}
1: h   : 97
2: i   : 1
  Stack
   3007
   98 ‘b’
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 97
2: i    : 1
  Stack
   3105
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0            // Push 'this'
 1: getfield   #2      // Push 'hash'
 4: istore_1           // reg1 = h = hash
 5: iload_1            // Push 'h'
 6: ifne       43      // early exit if hash!=0
 9: iconst_0           // 0
10: istore_2           // reg2 = i = 0
// LOOP
11: iload_2            // Push i
12: aload_0            // Push 'this'
13: getfield   #3      // 'value'
16: arraylength        // value.length
17: if_icmpge 38       // Normal exit if i>=value.length
20: bipush     31      // 31
22: iload_1            // h
23: imul               // h*31
24: aload_0            // Push 'this'
25: getfield   #3      // 'value'
28: iload_2            // Push 'i'
29: caload             // value[i]
30: iadd               // h*31 + value[i]
31: istore_1           // h = h*31 + value[i]
32: iinc 2, 1          // reg2+=1; i++
35: goto       11      // Loop repeats

38: aload_0            // Push 'this'
39: iload_1            // Push 'h'
40: putfield   #2      // hash = h;

43: iload_1            // Push 'h'
44: ireturn            // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 1
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0            // Push 'this'
 1: getfield   #2      // Push 'hash'
 4: istore_1           // reg1 = h = hash
 5: iload_1            // Push 'h'
 6: ifne       43      // early exit if hash!=0
 9: iconst_0           // 0
10: istore_2           // reg2 = i = 0
// LOOP
11: iload_2            // Push i
12: aload_0            // Push 'this'
13: getfield   #3      // 'value'
16: arraylength        // value.length
17: if_icmpge 38       // Normal exit if i>=value.length
20: bipush     31      // 31
22: iload_1            // h
23: imul               // h*31
24: aload_0            // Push 'this'
25: getfield   #3      // 'value'
28: iload_2            // Push 'i'
29: caload             // value[i]
30: iadd               // h*31 + value[i]
31: istore_1           // h = h*31 + value[i]
32: iinc 2, 1          // reg2+=1; i++
35: goto       11      // Loop repeats

38: aload_0            // Push 'this'
39: iload_1            // Push 'h'
40: putfield   #2      // hash = h;

43: iload_1            // Push 'h'
44: ireturn            // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   2
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 2
  Stack
   2
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield    #2   // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne        43   // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield    #3   // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush      31   // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield    #3   // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto        11   // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield    #2   // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   2
   [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 2
  Stack
   2
   3
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 2
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   31
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 2
  Stack
   31
   3105
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   96255
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 3105
2: i    : 2
  Stack
   96255
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge  38    // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   96255
   [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   96255
   [abc]
   2
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   96255
   99 'c'
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  3105
2: i    :  2
  Stack
   96354
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 96354
2: i    : 2
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0         // Push 'this'
 1: getfield   #2   // Push 'hash'
 4: istore_1        // reg1 = h = hash
 5: iload_1         // Push 'h'
 6: ifne       43   // early exit if hash!=0
 9: iconst_0        // 0
10: istore_2        // reg2 = i = 0
// LOOP
11: iload_2         // Push i
12: aload_0         // Push 'this'
13: getfield   #3   // 'value'
16: arraylength     // value.length
17: if_icmpge 38    // Normal exit if i>=value.length
20: bipush     31   // 31
22: iload_1         // h
23: imul            // h*31
24: aload_0         // Push 'this'
25: getfield   #3   // 'value'
28: iload_2         // Push 'i'
29: caload          // value[i]
30: iadd            // h*31 + value[i]
31: istore_1        // h = h*31 + value[i]
32: iinc 2, 1       // reg2+=1; i++
35: goto       11   // Loop repeats

38: aload_0         // Push 'this'
39: iload_1         // Push 'h'
40: putfield   #2   // hash = h;

43: iload_1         // Push 'h'
44: ireturn         // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
➡
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
  0: aload_0          // Push 'this'
  1: getfield   #2    // Push 'hash'
  4: istore_1         // reg1 = h = hash
  5: iload_1          // Push 'h'
  6: ifne       43    // early exit if hash!=0
  9: iconst_0         // 0
 10: istore_2         // reg2 = i = 0
// LOOP
 11: iload_2          // Push i
 12: aload_0          // Push 'this'
 13: getfield   #3    // 'value'
 16: arraylength      // value.length
 17: if_icmpge 38     // Normal exit if i>=value.length
 20: bipush     31    // 31
 22: iload_1          // h
 23: imul             // h*31
 24: aload_0          // Push 'this'
 25: getfield   #3    // 'value'
 28: iload_2          // Push 'i'
 29: caload           // value[i]
 30: iadd             // h*31 + value[i]
 31: istore_1         // h = h*31 + value[i]
 32: iinc 2, 1        // reg2+=1; i++
→35: goto       11    // Loop repeats

 38: aload_0          // Push 'this'
 39: iload_1          // Push 'h'
 40: putfield   #2    // hash = h;

 43: iload_1          // Push 'h'
 44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
   3
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0             // Push 'this'
 1: getfield    #2      // Push 'hash'
 4: istore_1            // reg1 = h = hash
 5: iload_1             // Push 'h'
 6: ifne        43      // early exit if hash!=0
 9: iconst_0            // 0
10: istore_2            // reg2 = i = 0
// LOOP
11: iload_2             // Push i
12: aload_0             // Push 'this'
13: getfield    #3      // 'value'
16: arraylength         // value.length
17: if_icmpge 38        // Normal exit if i>=value.length
20: bipush      31      // 31
22: iload_1             // h
23: imul                // h*31
24: aload_0             // Push 'this'
25: getfield    #3      // 'value'
28: iload_2             // Push 'i'
29: caload              // value[i]
30: iadd                // h*31 + value[i]
31: istore_1            // h = h*31 + value[i]
32: iinc 2, 1           // reg2+=1; i++
35: goto        11      // Loop repeats

38: aload_0             // Push 'this'
39: iload_1             // Push 'h'
40: putfield    #2      // hash = h;

43: iload_1             // Push 'h'
44: ireturn             // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 96354
2: i    : 3
  Stack
   3
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0           // Push 'this'
 1: getfield    #2    // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne        43    // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield    #3    // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush      31    // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield    #3    // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto        11    // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield    #2    // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
    3
    [abc]
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
   3
   3
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 96354
2: i    : 3
  Stack
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge  38    // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=0; value=[abc]}
1: h    : 96354
2: i    : 3
  Stack
   String{hash=0; value=[abc]}
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0             // Push 'this'
 1: getfield    #2      // Push 'hash'
 4: istore_1            // reg1 = h = hash
 5: iload_1             // Push 'h'
 6: ifne        43      // early exit if hash!=0
 9: iconst_0            // 0
10: istore_2            // reg2 = i = 0
// LOOP
11: iload_2             // Push i
12: aload_0             // Push 'this'
13: getfield    #3      // 'value'
16: arraylength         // value.length
17: if_icmpge 38        // Normal exit if i>=value.length
20: bipush      31      // 31
22: iload_1             // h
23: imul                // h*31
24: aload_0             // Push 'this'
25: getfield    #3      // 'value'
28: iload_2             // Push 'i'
29: caload              // value[i]
30: iadd                // h*31 + value[i]
31: istore_1            // h = h*31 + value[i]
32: iinc 2, 1           // reg2+=1; i++
35: goto        11      // Loop repeats

38: aload_0             // Push 'this'
39: iload_1             // Push 'h'
40: putfield    #2      // hash = h;

43: iload_1             // Push 'h'
44: ireturn             // return h;
```

Interpreter

```
  Locals
0:this  :  String{hash=0; value=[abc]}
1: h    :  96354
2: i    :  3
  Stack
   String{hash=0; value=[abc]}
   96354
```

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter

```
  Locals
0:this  : String{hash=96354; value=...
1: h    : 96354
2: i    : 3
  Stack
```

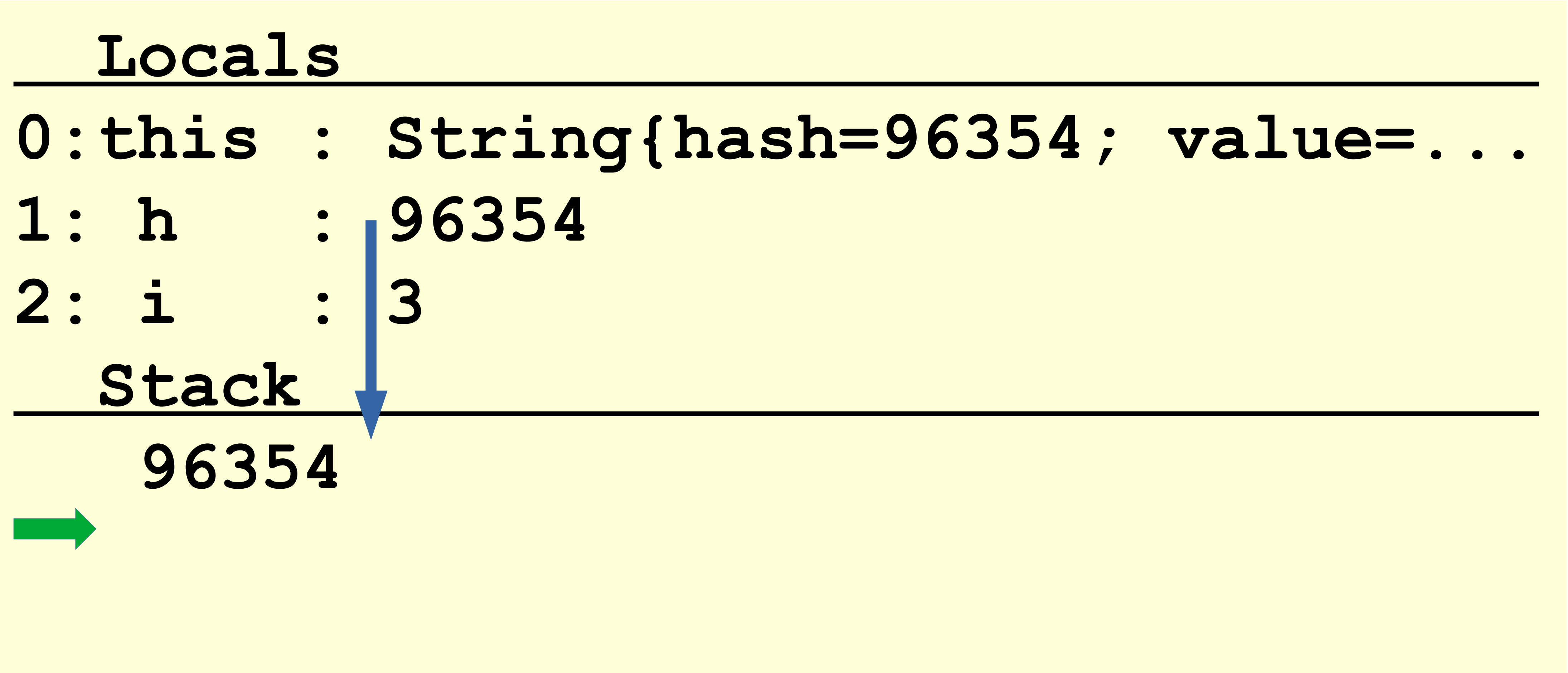
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0          // Push 'this'
 1: getfield   #2    // Push 'hash'
 4: istore_1         // reg1 = h = hash
 5: iload_1          // Push 'h'
 6: ifne       43    // early exit if hash!=0
 9: iconst_0         // 0
10: istore_2         // reg2 = i = 0
// LOOP
11: iload_2          // Push i
12: aload_0          // Push 'this'
13: getfield   #3    // 'value'
16: arraylength      // value.length
17: if_icmpge 38     // Normal exit if i>=value.length
20: bipush     31    // 31
22: iload_1          // h
23: imul             // h*31
24: aload_0          // Push 'this'
25: getfield   #3    // 'value'
28: iload_2          // Push 'i'
29: caload           // value[i]
30: iadd             // h*31 + value[i]
31: istore_1         // h = h*31 + value[i]
32: iinc 2, 1        // reg2+=1; i++
35: goto       11    // Loop repeats

38: aload_0          // Push 'this'
39: iload_1          // Push 'h'
40: putfield   #2    // hash = h;

43: iload_1          // Push 'h'
44: ireturn          // return h;
```

Interpreter



```
  Locals
0:this  : String{hash=96354; value=...
1: h    : 96354
2: i    : 3
  Stack
   96354
```

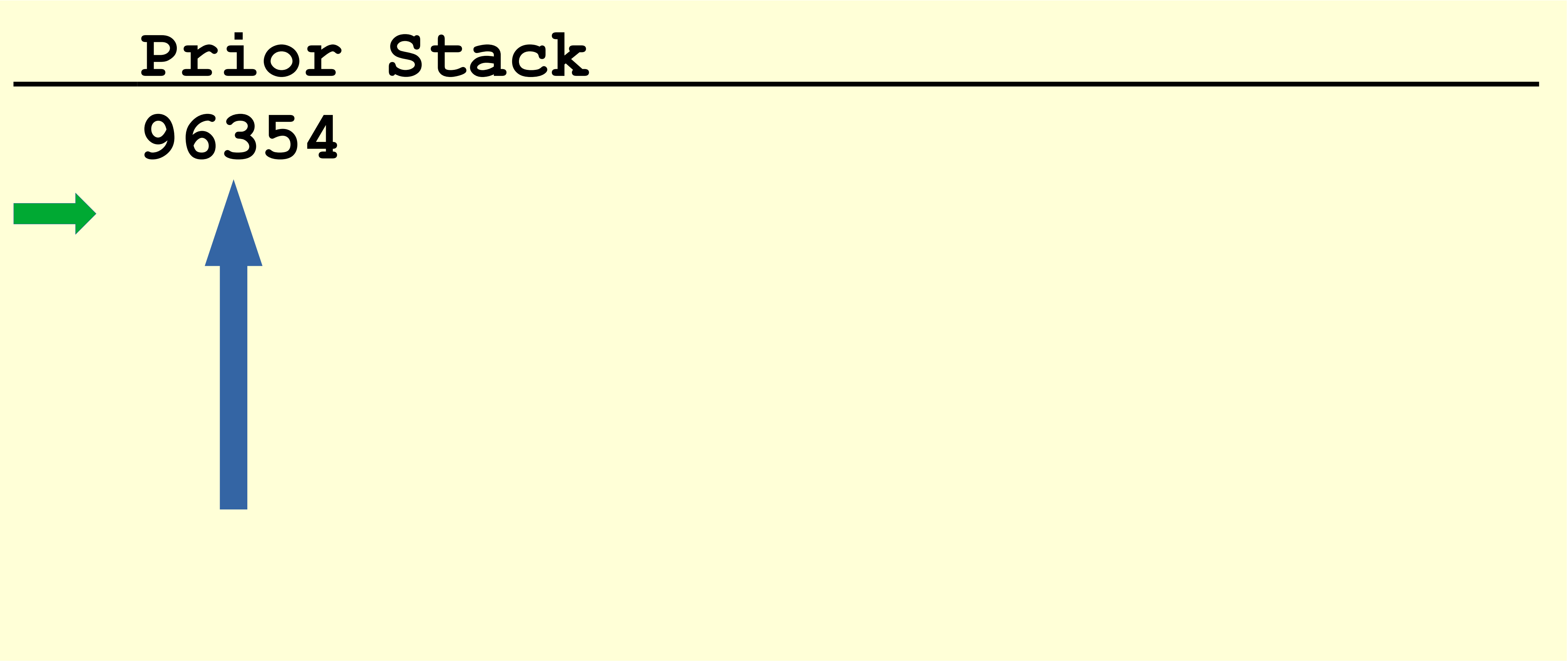
```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h
}
```

```
 0: aload_0           // Push 'this'
 1: getfield   #2     // Push 'hash'
 4: istore_1          // reg1 = h = hash
 5: iload_1           // Push 'h'
 6: ifne       43     // early exit if hash!=0
 9: iconst_0          // 0
10: istore_2          // reg2 = i = 0
// LOOP
11: iload_2           // Push i
12: aload_0           // Push 'this'
13: getfield   #3     // 'value'
16: arraylength       // value.length
17: if_icmpge 38      // Normal exit if i>=value.length
20: bipush     31     // 31
22: iload_1           // h
23: imul              // h*31
24: aload_0           // Push 'this'
25: getfield   #3     // 'value'
28: iload_2           // Push 'i'
29: caload            // value[i]
30: iadd              // h*31 + value[i]
31: istore_1          // h = h*31 + value[i]
32: iinc 2, 1         // reg2+=1; i++
35: goto       11     // Loop repeats

38: aload_0           // Push 'this'
39: iload_1           // Push 'h'
40: putfield   #2     // hash = h;

43: iload_1           // Push 'h'
44: ireturn           // return h;
```

Interpreter



Prior Stack

96354

```
public int hashCode() {
  int h = hash;
  if( h == 0 ) {
    for( int i=0; i<value.length; i++ )
      h = 31 * h + value[i];
    hash = h;
  }
  return h;
}
```

```
 0: aload_0         // Push 'this'
 1: getfield   #2   // Push 'hash'
 4: istore_1        // reg1 = h = hash
 5: iload_1         // Push 'h'
 6: ifne       43   // early exit if hash!=0
 9: iconst_0        // 0
10: istore_2        // reg2 = i = 0
// LOOP
11: iload_2         // Push i
12: aload_0         // Push 'this'
13: getfield   #3   // 'value'
16: arraylength     // value.length
17: if_icmpge 38    // Normal exit if i>=value.length
20: bipush     31   // 31
22: iload_1         // h
23: imul            // h*31
24: aload_0         // Push 'this'
25: getfield   #3   // 'value'
28: iload_2         // Push 'i'
29: caload          // value[i]
30: iadd            // h*31 + value[i]
31: istore_1        // h = h*31 + value[i]
32: iinc 2, 1       // reg2+=1; i++
35: goto       11   // Loop repeats

38: aload_0         // Push 'this'
39: iload_1         // Push 'h'
40: putfield   #2   // hash = h;

43: iload_1         // Push 'h'
44: ireturn         // return h;
```

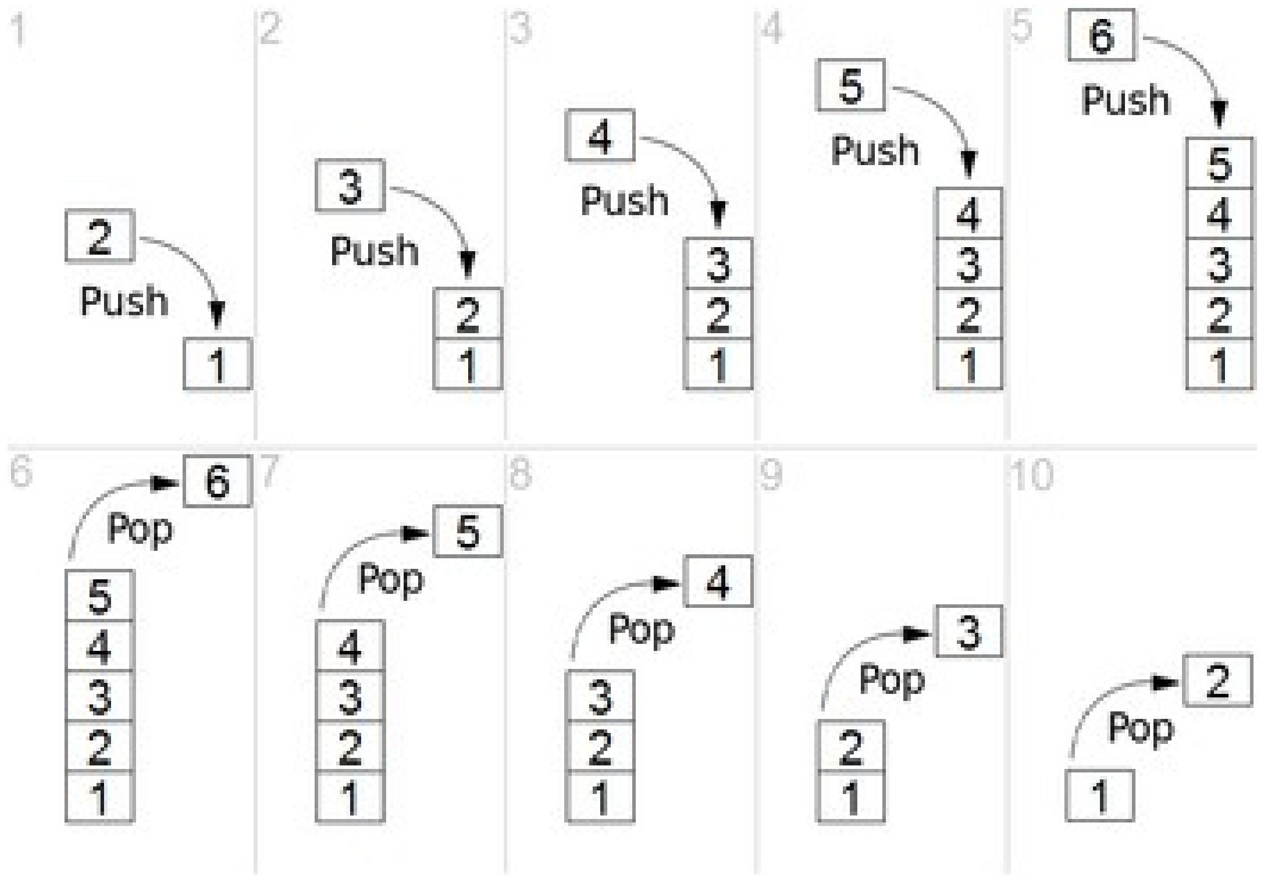
[redacted]

Interpreter

Bytecodes are hard to work with:

- Very nit-picky & detailed
- Conversion from javac to bytecodes leaves a lot to be desired
- (Re)loading the same thing over and over
- Lots of stack motion

Lots of room for improvement!



[redacted]

Bytecode Snippets: “Do”

Bytecodes do ‘stack-like’ things

- Semantics: pop,pop,imul,push
- Generally top-of-stack already in a register

imul X86 assembly, TOS in RAX already:

```
// rax=97 ‘a’
// Stack: 31
pop  rdx // rdx=31
imul     // rax=3007
jmp  next
```

aload_0: Locals base in RDI already:

```
push rax // prior TOS
mov  rax,[rdi+8*0]
jmp  next
```

getfield#2: Not so easy!

- Check bytecode is “resolved”; slow-path if not
- Check for null & throw
- Check for jvmti, debug checks
- Fast case: about ~30 ops

Bytecode Dispatch Overhead

“See” next bytecode

- Fetch bytecode, bump
- Jump to “Do” handler

```
// rsi: bytecode pointer
ld1 rbx,[rsi+off]     // fetch bytecode
add rsi,off           // advance pointer
jmp [0xtable + rbx*8] // jump to “Do” snippet
```

Load is 3+ clks, Jump is 30+ clks

- Dwarfs all the simple “Do” handlers
- Dispatch is about 50% of overhead

Template JIT:

- Copies the “Do” handlers back-to-back
- No “See” dispatch during execution

Summary

An Execution Model describes program semantics

- Interpretation is just one way to execute
- Easy to build and get started with

The **Interpreter** has serious overheads & is slow

- Bytecodes are hard to work with!
- Lots of room for improvement!

Can’t we do better?

Part 2: Moving To An Intermediate Representation