

Surprize Snippets

John Motil
Professor Emeritus
California State University
Northridge

Many Surprises: to motivate **curiosity**

Sphere Surprise

Surprise in Time

Maximum Surprise

Surprise of Sorts

Tree Surprise

Gambling Surprise

Sphere Surprise

```
// Volume of a sphere ?
```

```
double radius, volume;
```

```
radius = 1.0; // a unit sphere
```

```
volume = (4 / 3) * 3.14159 * radius * radius * radius;
```

```
System.out.print (volume);
```

`volume = (4 / 3) * 3.14159 * radius * radius * radius;`

$$(4 / 3) = 1$$

$$(4.0 / 3.0) = 1.3333\dots$$

BIG Error

BUT no error message!

Be Aware!

Beware!

Java!

Surprize in Time

```
// Does tell military time
int hour, mins, milTime;
milTime = 1234;
milTime = 0432; //try it!

hour = milTime / 100;
mins = milTime % 100;

System.out.print ("Time is ");
System.out.print (mins);
System.out.print (" minutes after ");
System.out.print (hour);
```

It converts 1234 into
"34 minutes after 12",

BUT

it converts 0432 into
"2 minutes after 82"

WHY?

WHY

it converts 0432 into
"2 minutes after 82"

because

The leading zero in **0432**
causes the number to be interpreted as
octal (base 8) in Java !

$$\begin{aligned} \mathbf{432} &= \text{in base 8} \\ &= \mathbf{4} * 8^2 + \mathbf{3} * 8 + \mathbf{2} \\ &= 282 \text{ (in decimal, base ten)} \end{aligned}$$

Be Aware
Beware
Java!

Maximum Surprise:

Given an algorithm, or code, computing the maximum max of a number num = 100 values

What is the number of times that max gets updated?

```
Input num  
Input max
```

```
Repeat num times
```

```
    Input value
```

```
        If (max < value) then  
            Set max = value  
            -- Output max  
        EndIf
```

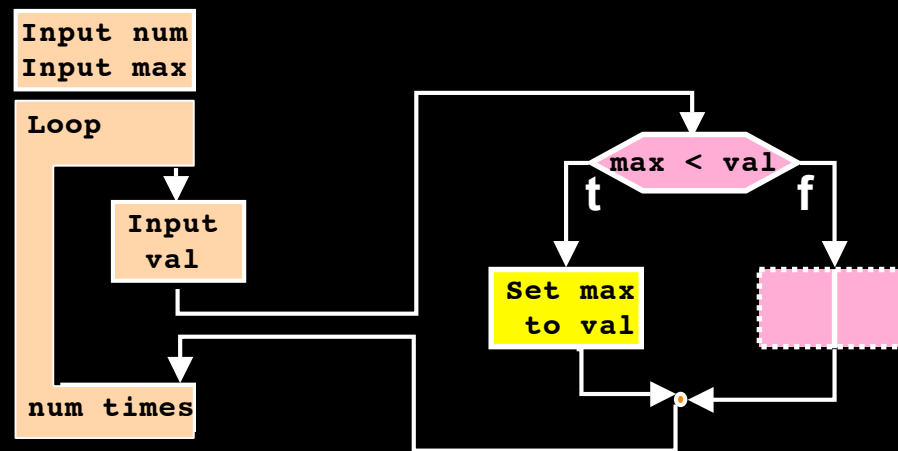
```
EndRepeat
```

```
Output max
```

```
num = 100;  
max = Math.random();  
  
for (i = 0; i < num; i++) {  
    value = Math.random();  
  
    if (max < value) {  
        max = value;  
        // output(max);  
    } // EndIf  
  
} // end for  
  
System.out.println (max);
```

Maximum Surprise: A graphic view

What is the number of **times** that max gets **updated**?



The answer is:

4.2

**Max gets updated 4.2 times out of 100
(approximately)**

This can be verified in many ways:

- 1. A Simple simulation**
- 2. Harmonic Series**
- 3. Sterlings result**

Try it !

Maximum Surprise **code** and some typical **runs**

```
1 // Maximum Surprise
2
3 int num = 100;
4 double value, max;
5
6 max = Math.random();
7
8 for (int i = 0; i < num; i++) {
9
10     value = Math.random();
11
12     if (max < value) {
13         max = value;
14         System.out.println(max);
15     }//EndIf
16
17 }//end for 100 random values
18
19 System.out.print ("Max is ");
20 System.out.println (max);
```

```
BEGIN: snippet
0.6772467155179934
0.806903062463738
0.9802621483505422
0.9861080617295355
0.9979460909135723
Max is 0.9979460909135723
END: snippet
jj$ BEGIN: snippet
0.542907745456997
0.9856007278142449
Max is 0.9856007278142449
END: snippet
jj$ BEGIN: snippet
0.3216707021154547
0.9014576249059495
0.9727187697997673
0.9908004149833322
Max is 0.9908004149833322
END: snippet
jj$ BEGIN: snippet
0.92189856742096
0.9608099131542527
0.9939774064867105
Max is 0.9939774064867105
END: snippet
```

A **Bologna** Surprise:

A solution to this Maximum Surprise
is that the number of updates of max is:

$$1/2 + 1/3 + 1/4 + 1/5 + \dots 1/n$$

which is related to the Harmonic Series.

A surprise is that in **1647**

Pietro Mengoli of **Bologna**

proved this series **diverges**.

Another surprise is that I found

20 other proofs of the divergence of this series!

Problems on Maximum Surprise

Find the number of updates for 1,000 random values, then again for 10,000, 100,000, 1 million, 2 million. (but guess first, and be surprised over and over). Do a few runs of each to see the variation.

For the case of 100 random values find the average number of updates (looping over thousands of times).

Create a histogram (for 100 random values) to show the percent of the time that n updates occur.

Relate the number of updates to the Harmonic Series
 $1 + 1/2 + 1/3 + 1/4 + 1/5. \dots$
and write code to check this. Why the weird error?

Surprize of **Sorts**

Consider the Bubble Sort (or SwapSort)

```
// First Swap Sort
i = 0;
while (i < size) {

    for (j = 0; j < last; j++) {
        if (a[j] < a[j+1]) {
            // swap adjacent items
            temp    = a[ j ];
            a[ j ] = a[j+1];
            a[j+1] = temp;

        } //end if swap
    } //end for j

    i++;
} //end while i < size
```

It is often modified as follows:

Modified Sort

This version stops when the array is sorted

```
// Second Swap Sort
done = false;
while ( ! done ) {
    done = true;

    for (j = 0; j < last; j++) {
        if (a[j] < a[j+1]) {
            // swap adjacent items
            temp    = a[ j ];
            a[ j ] = a[j+1];
            a[j+1] = temp;
            done = false;
        } //end if swap
    } //end for j
} //end while not done
```

Compare these sorts, side by side

```
// First Swap Sort
i = 0;
while (i < size) {

    for (j = 0; j < last; j++) {
        if (a[j] < a[j+1]) {
            // swap adjacent items
            temp = a[ j ];
            a[ j ] = a[j+1];
            a[j+1] = temp;

        } //end if swap
    } //end for j

    i++;
} //end while i < size
```

```
// Second Swap Sort
done = false;
while ( ! done ) {
    done = true;

    for (j = 0; j < last; j++) {
        if (a[j] < a[j+1]) {
            // swap adjacent items
            temp = a[ j ];
            a[ j ] = a[j+1];
            a[j+1] = temp;
            done = false;
        } //end if swap
    } //end for j

} //end while not done
```

Guess at the percent speed increase, **psi**;
by what percent is the second one faster than the first one?

The answer is:

- 4.5

The percent increase in speed is **NEGATIVE**

and small and approximate,

depending on the computer, language, etc

The second one is always **slower** than the first !

WHY ?

The extra code which was inserted to speed it up, actually took more time to run, so canceling out the benefit in speed.

Problems on **Sorts**

NonLogical SwapSort

Modify the logical swapSort by replacing the boolean box done with an integer box swapCount, which counts the number of times that a swap occurred within a pass. Compare the time. Output this swapCount after each pass to see how it varies.

YoYo Sort

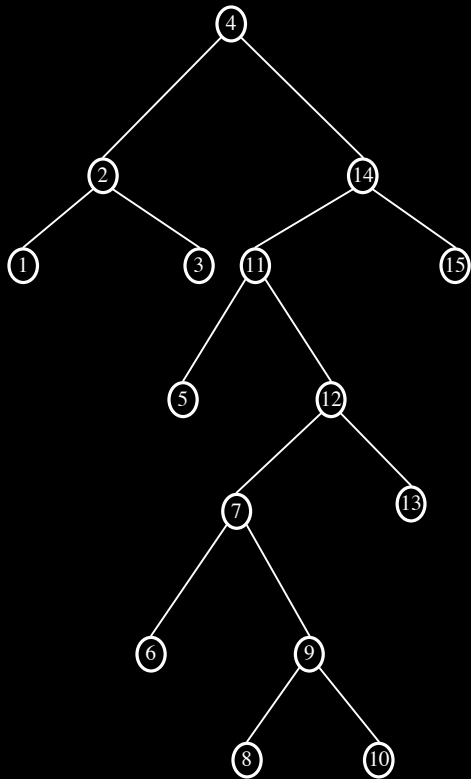
Modify the second swapSort algorithm by alternating the direction of the passes; for example first pass from 0 to m, then pass from m down to 0, and keep alternating this direction. Do you expect this sort to be much faster? Try it.

Shell Sort

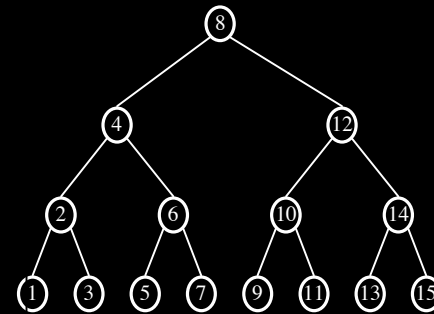
Swapsort compares adjacent values; modify it to compare values a distance $n/2$ apart, and then $n/4$ apart, etc. This sort will be significantly faster.

Tree Surprise

Given a number of elements (say 1000) arranged as a binary search tree, or BST, what is the average height of the tree (if random elements are inserted).



The two BS tree examples show 15 elements; (each left “child” is smaller than the parent, and each right one is larger than the parent). The tree at the left has height 6. The tree at the right has height 3. The worst tree has a height of 15.



For 1000 elements
the best tree height is 10.
the worst tree height is 1000;

What is your guess for the average?

The answer is:

The average height of a tree
of 1000 random values is:

21

approximately

**This can be verified with
a Simple simulation**

Try it !

Gambling Surprise

Suppose that you have 5 dollars to gamble and that the house has twice as much, \$10.

You play a dollar on each game, and your probability of winning is 0.5 (coin toss).

You play until one of you wins all of it.
What is your probability of winning it all?

The answer is:

1/3

**You have only a 1 in 3 chance of winning it all !
(even though you have a 1 in 2 chance to win each game).**

**This can be verified in many ways:
a simple simulation gives insight**

Try it !

Gambling does not pay

(even if odds are in your favor!)

Gamble Surprise code and some typical runs

```
1
2 // GAMBLING SURPRIZE:
3 // Does show that gambling does not pay
4 // even if the odds are in your favor !
5
6 double rand; // random number
7 int amount; // initial amount
8 double winProb; // win probability
9
10 // INPUT: experiment with following
11 amount = 5; // whatever
12 winProb = 0.50 ; // most fair
13 // winProb = 0.493; // dice craps
14 // winProb = 0.525; // impossible
15
16 // GAMBLE: until someone goes broke
17 while ((amount > 0) && (amount < 15)) {
18 // Exit when amount is 0 or 15
19 rand = Math.random();
20 if (rand < winProb) {
21 amount ++; // Win a dollar
22 }else{
23 amount --; // Lose a dollar
24 }//end if
25 System.out.print (amount + " ");
26 }//end while nobody is broke
27
28 // REPORT: the final result
29 if (amount == 0) {
30 System.out.println ("You Lose all!");
31 }else{ // (amount == 15)
32 System.out.println ("YOU WIN ALL !");
33 }//end if
```

```
jj$ BEGIN: snippet
4 5 4 3 4 3 2 3 2 1 0 You Lose all!
END: snippet
jj$ BEGIN: snippet
4 5 4 5 6 7 6 7 8 7 8 7 6 7 6 5 4 3 4 5 6 5 4 3 4 5 4 3
4 5 6 7 8 7 6 7 6 5 6 7 8 7 6 5 6 5 6 7 8 7 8 7 6 7 8 7
6 7 6 5 4 5 6 7 6 7 6 5 6 5 6 7 6 5 4 3 2 1 0 You Lose
all!
END: snippet
jj$ BEGIN: snippet
6 7 6 7 8 9 8 7 8 7 8 7 6 7 8 9 8 7 8 7 6 7 8 7 8 9 10
11 12 13 14 15 YOU WIN ALL !
END: snippet
jj$ BEGIN: snippet
4 3 2 3 2 3 2 3 4 5 6 7 6 5 6 5 4 5 6 7 8 7 8 9 10 11
10 9 10 11 10 11 10 11 10 11 12 11 10 11 10 9 10 11 10 9 10 9
8 7 8 9 10 11 10 11 10 9 10 11 12 11 12 13 12 11 10 11
12 13 14 15 YOU WIN ALL !
END: snippet
jj$ BEGIN: snippet
6 5 4 5 4 5 4 3 4 5 4 5 4 3 2 3 4 3 4 5 4 5 4 5 6 7 6 7
6 7 8 7 6 5 6 7 8 9 8 7 6 5 6 5 4 5 6 7 8 7 6 7 8 7 8 9
10 9 10 9 8 7 8 7 8 9 8 7 8 9 10 9 8 7 8 9 8 7 6 7 6 5
4 5 6 5 4 3 2 1 0 You Lose all!
END: snippet
jj$ BEGIN: snippet
4 5 6 7 6 7 8 9 8 9 10 11 10 9 8 9 8 9 10 11 10 11 10 9
8 7 6 5 4 5 6 5 6 5 4 5 4 5 4 5 6 7 8 7 6 5 6 5 4 3 2 3
2 3 4 5 4 5 6 5 4 5 4 5 6 5 4 3 2 3 4 5 6 5 4 5 4 5 4 5
4 3 2 3 4 5 4 3 4 5 6 5 6 7 6 7 8 7 6 7 6 5 4 5 4 3 4 5
4 5 6 5 6 7 6 5 4 5 4 3 4 3 2 1 0 You Lose all!
END: snippet
jj$
```

Problems on Gambling Surprise

What is the average **length** of game. (Answer is 50)

What is your average **maximum** amount of your win (and why didn't you stop then?)

If the game involved is dice **craps** (with a probability of win of 0.493) what is your chance to win it all.

What probability of winning a game is needed for you to **win 50** percent of the games?

What if you **begin** with only 3 dollars?

What if the house has 3 times your amount?

4 times, 5 times, **n times**. Generalize.

Surprize Snippets

John Motil

Professor Emeritus

California State University, Northridge

More **surprizes** and nifty nonsense at:

`http://www.csun.edu/~jmotil/`