Algorithmic Thinking and Structured Programming (in Greenfoot)

Teachers:

Renske Smetsers-Weeda

Sjaak Smetsers

Ana Tanase

Today's Lesson plan (7)

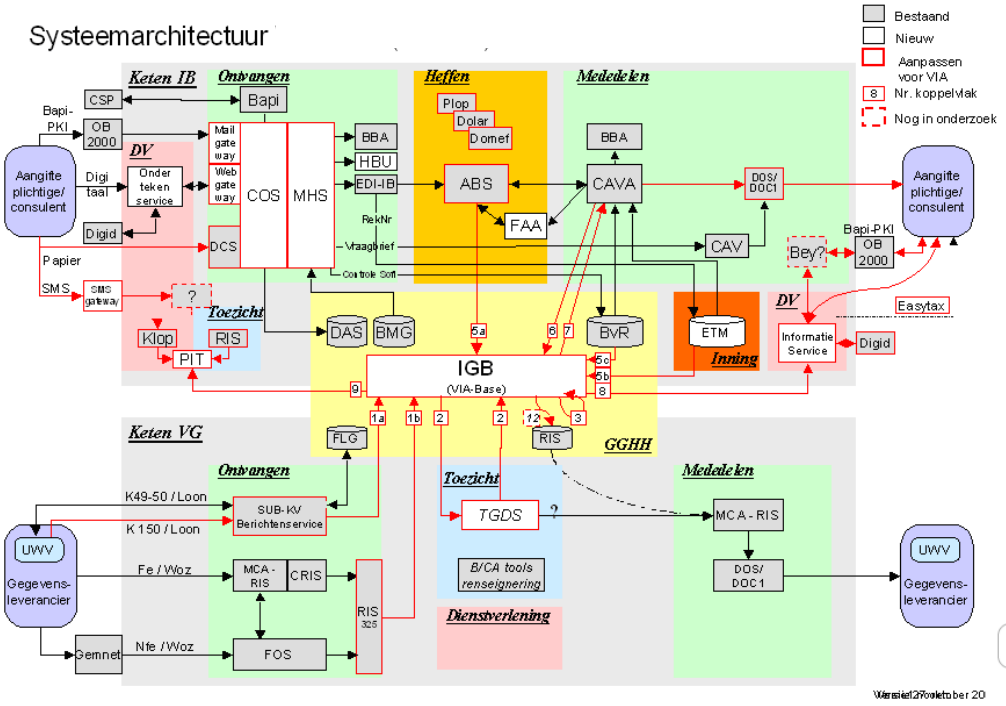
- Mid-Task

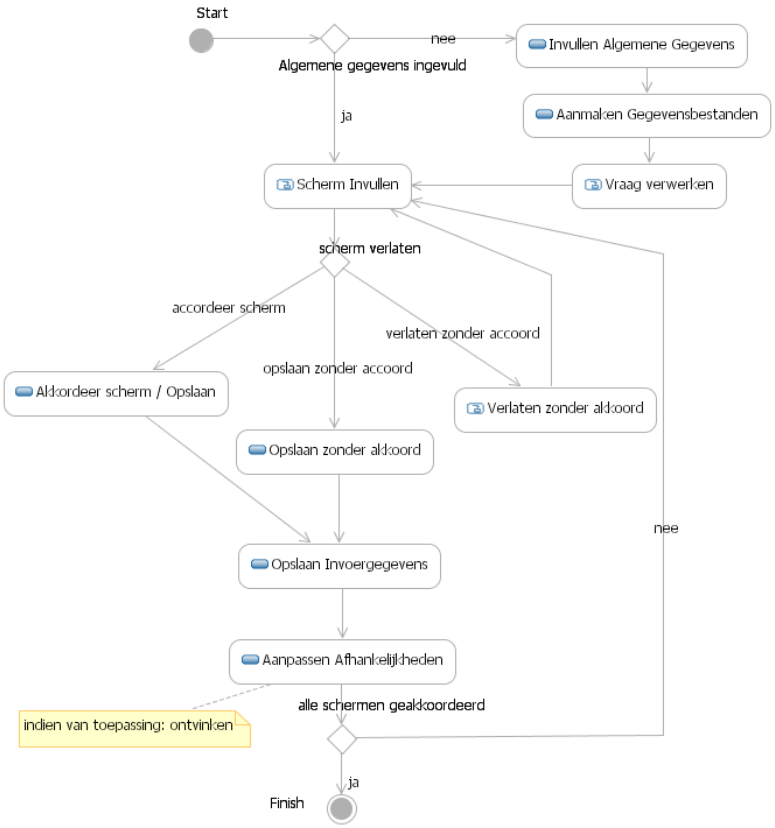
- Flowcharts (Real world example)
- Unplugged activity: sorting algorithms and efficiency
- Retrospective
- Theory: counter in while loop

- Assignments

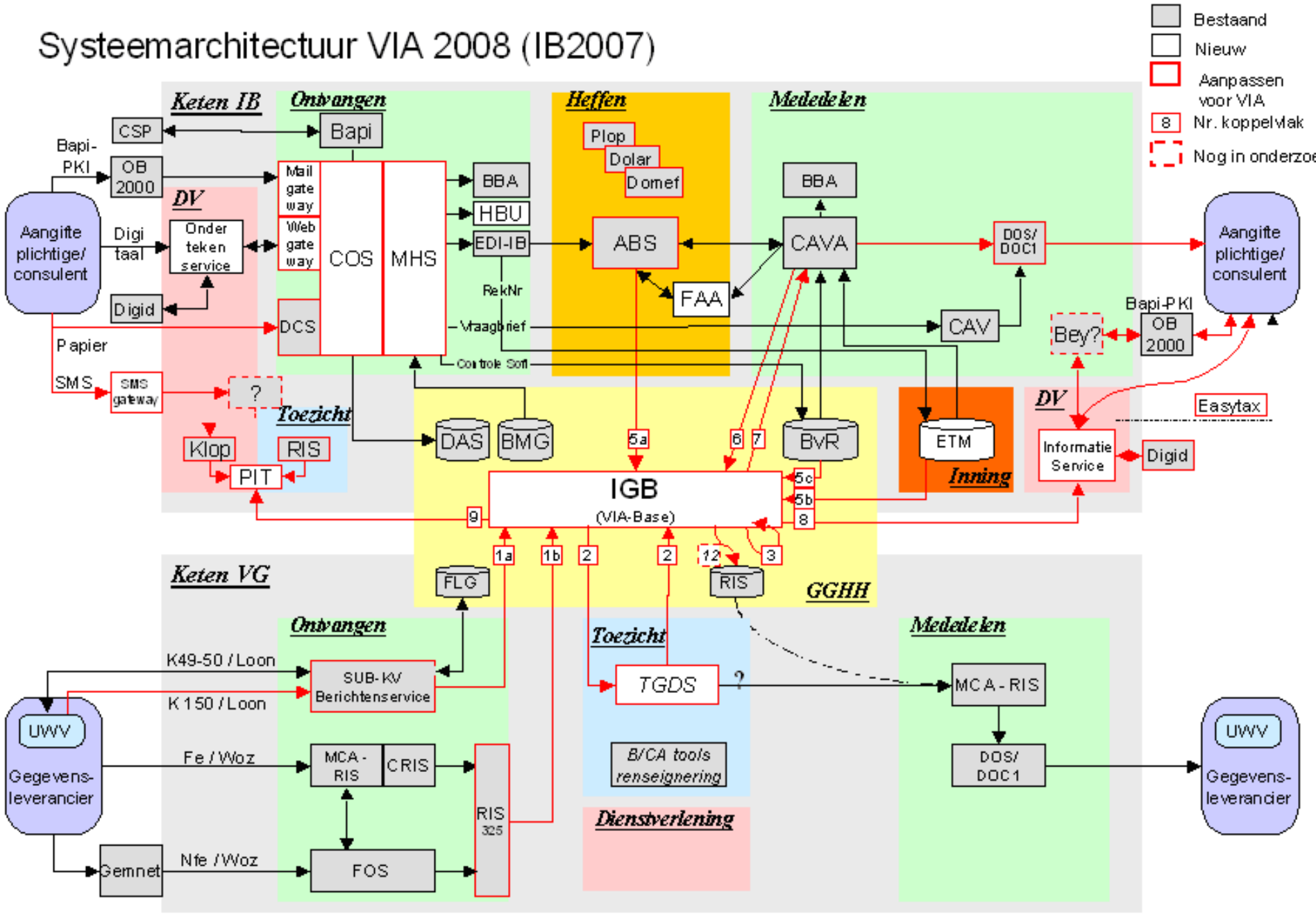
- Next week (Feb 5th): Quiz

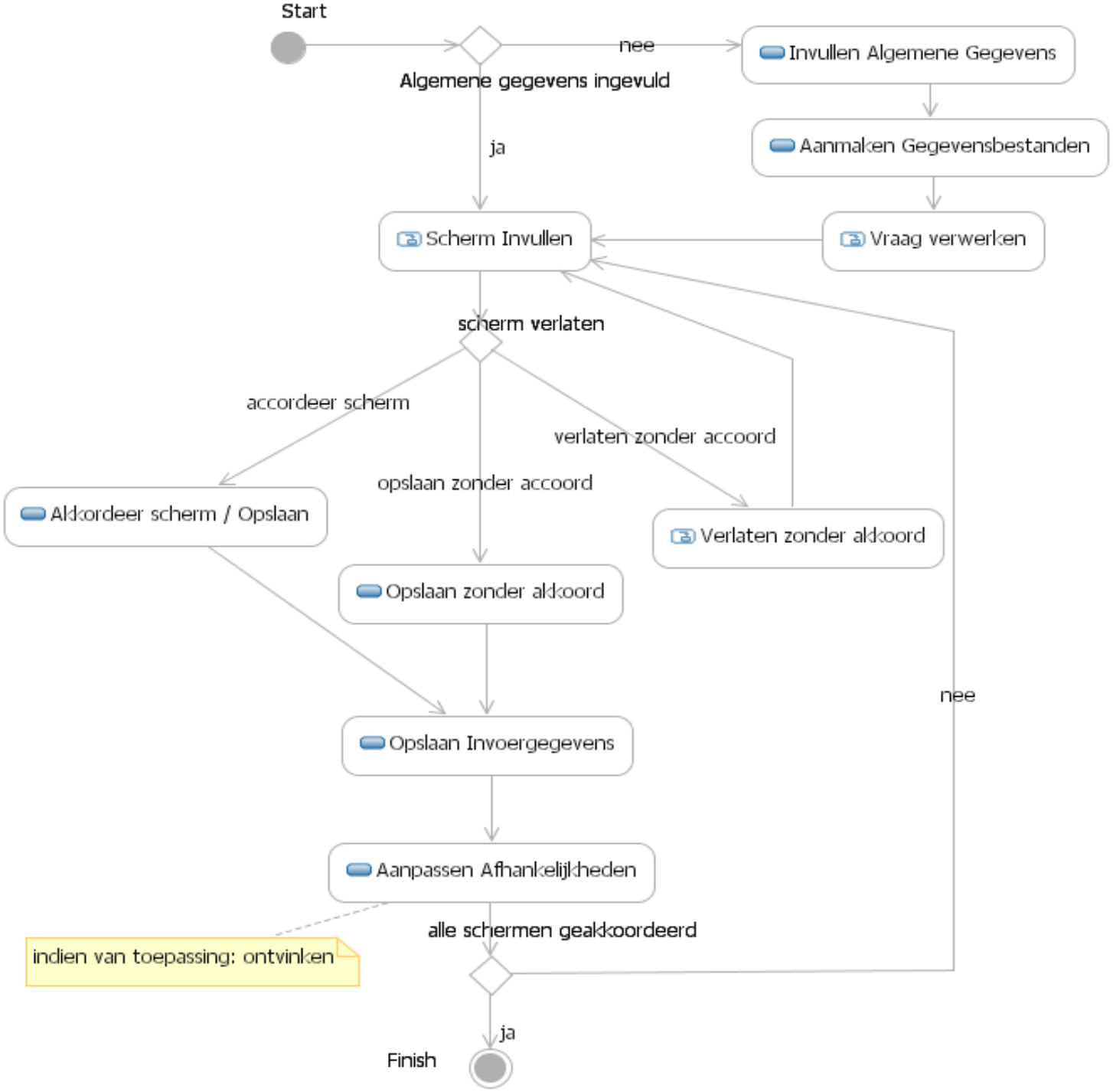
Flowcharts: Real world example





Systeemarchitectuur VIA 2008 (IB2007)

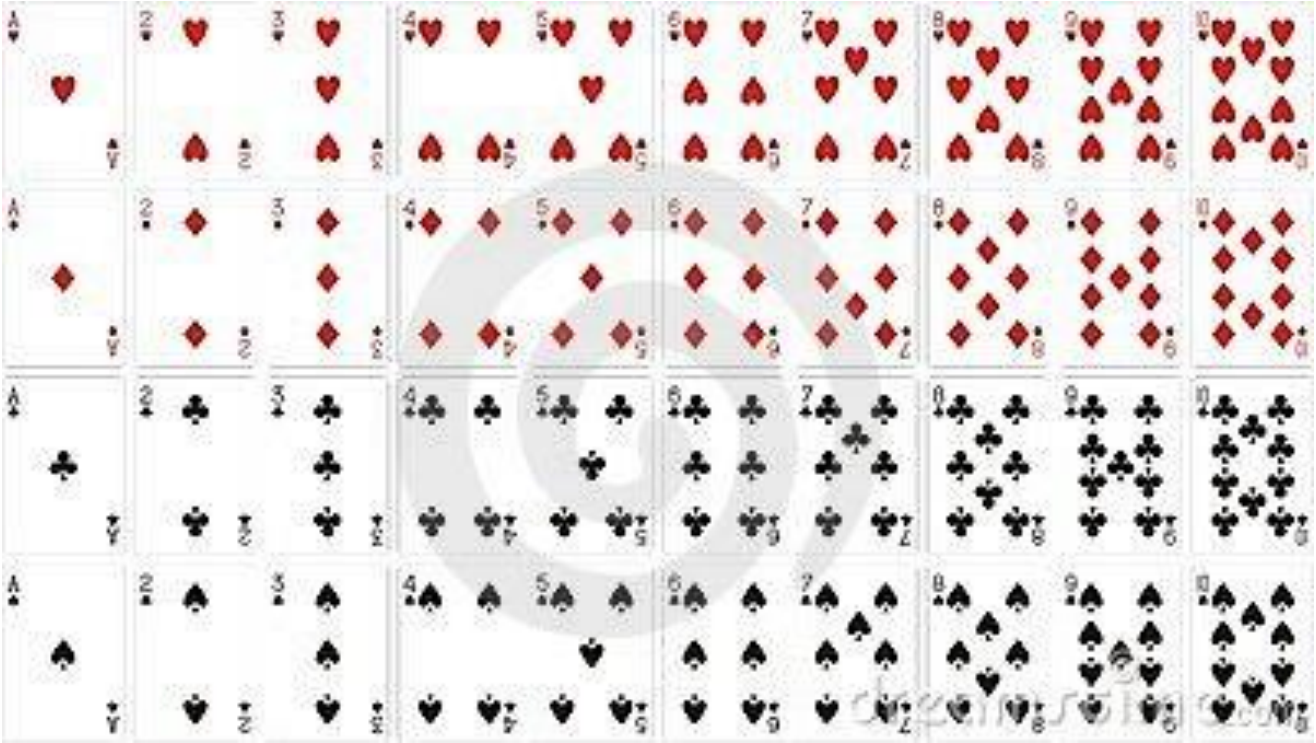



Start
Algemene gegevens ingevuld
nee
Invullen Algemene Gegevens
Aanmaken Gegevensbestanden
ja
Scherm Invullen
Vraag verwerken
scherm verlaten
accordeer scherm
verlaten zonder accoord
opslaan zonder accoord
Akkordeer scherm / Opslaan
Verlaten zonder akkoord
Opslaan zonder akkoord
nee
Opslaan Invoergegevens
Aanpassen Afhankelijkheden
alle schermen geakkoordeerd
indien van toepassing: ontvinken
ja
Finish

Unplugged

- Sorting algorithms and efficiency

Sort cards: Bogo Sort



Sort algorithms

Goal: Sort cups using only a balance

Sort algorithms (in pairs, 3 minutes)

- Goal: Sort cups using only a balance
  - order: lightest to heaviest
  - nr of steps?
- Describe an algorithm (with a flowchart) using basic instructions which a 4-year-old should be able to follow:
    - getCup (thirdCup)
    - determineLightestCup (thirdCup, seventhCup)

Sort algorithms: efficiency (2 minutes)

- Efficiency: Write down how many steps if you have:
  - 10 cups
  - 20 cups
  - 100 cups

Sort algorithms

- Share:
  - What did you come up with?
  - Efficiency

Quick sort: divide and conquer

1) Select a card at random
2) Divide collection into two groups:
    A) larger than selected card
    B) smaller than selected card
3) Give each pile of cards to another team
    & sit back and relax
4) Other teams repeat steps 1-3

When are we done?

Quick sort: divide and conquer

0) If you have 0 or 1 card, then STOP

1) Select a card at random

2) Divide collection into two groups:

    A) larger than selected card

    B) smaller than selected card

3) Give each pile of cards to another team

    Other teams repeat steps 1-3

Result: cards sorted from smallest to largest

Method: divide and conquer (recursive algorithm)

Quick sort summary

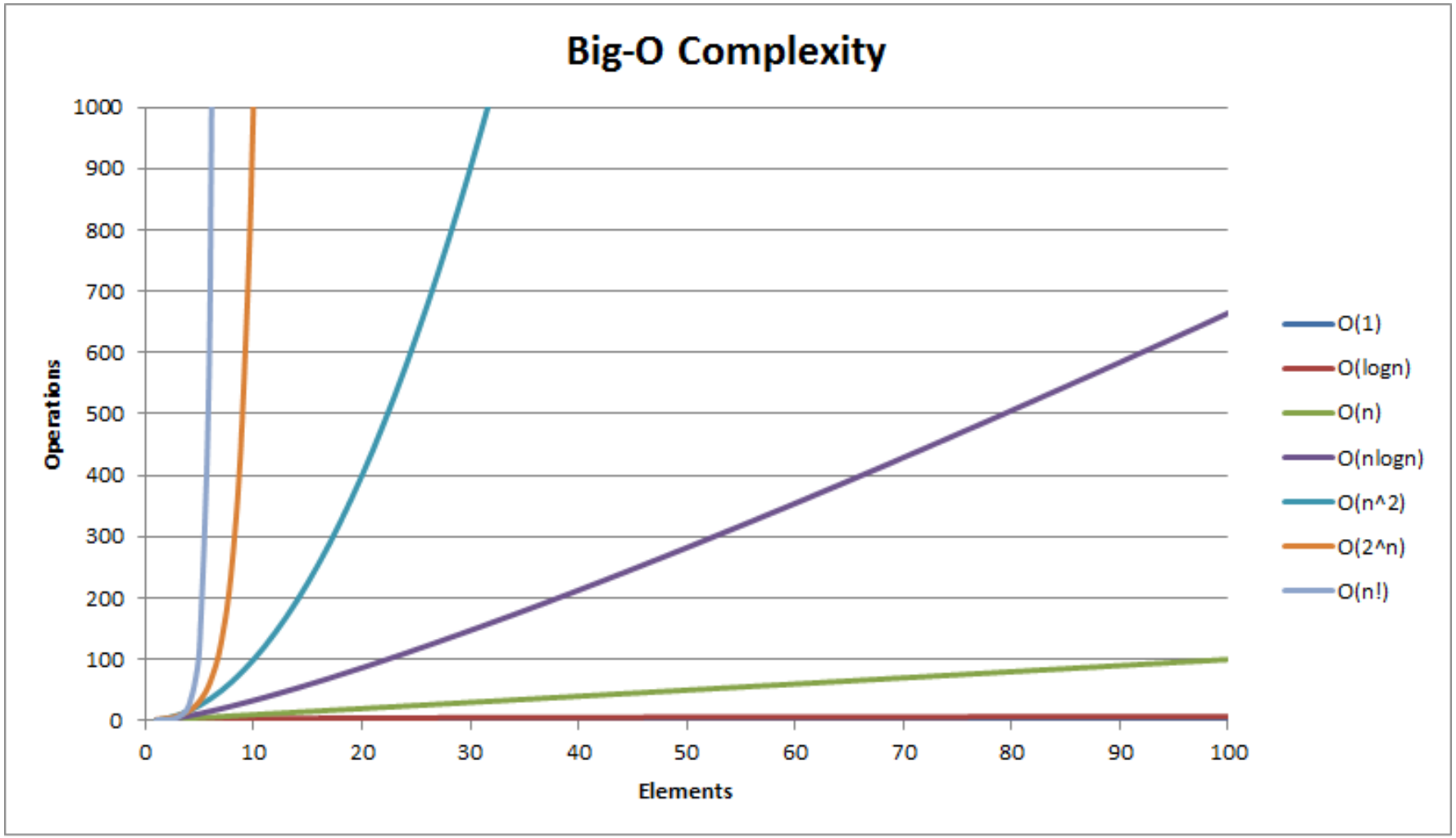
- Divide and conquer: Recursive programming
- Simple instructions
- Complexity n*log(n))

Growth Rates Compared:

	n=1	n=2	n=4	n=8	n=16	n=32
1	1	1	1	1	1	1
logn	0	1	2	3	4	5
n	1	2	4	8	16	32
nlogn	0	2	8	24	64	160
n^2	1	4	16	64	256	1024
n^3	1	8	64	512	4096	32768
2^n	2	4	16	256	65536	4294967296
$n!$	1	2	24	40320	20.9T	Don't ask!

Quick sort summary

- Complexity O(n*log(n)): purple curve



How much better is QuickSort?

https://www.youtube.com/watch?v=aXXWXz5rF64

Computational thinking

- **Working in a structured manner:**
  - Breaking problems down into subproblems
  - Design, solve and test solutions to subproblems
  - Combining these (sub)solutions to solve problem
- **Analyzing** the quality of a solution
- **Reflecting** about the solution chosen and proces
- **Generalizing** and re-use of existing solutions

Today's Lesson plan (7)

- Mid-Task
- Flowcharts (Real world example)
- Unplugged activity: algorithm efficiency

- Retrospective
- Theory: counter in while loop

- Assignments

- Next week (Feb 5th): Quiz

Retrospective

- Variables and Operators:
  - Assignment: =, +=, …
  - Arithmetic: +,-,*, ++, …
  - Comparisons: <, ==, <= …
- Tracing code

Variables and Values

- Assigning values

```
eggsPerBasket = 6;
totalEggs = eggsPerBasket + 3;
eggsPerBasket = eggsPerBasket - 2;
eggsPerBasket++; //increase value by 1
```

- Comparing values

```
if ( totalEggs <= 6 ){
…
}
```

Variable Swapping strategy

```
int a = 12;
int b = 4;

int temp = a;
a = b;
b = temp;
```

CODE	VALUE OF a	VALUE OF b	VALUE of temp

`int a = 12;`
`int b = 4;`
`int temp = a;`
	12	4	12

`a = b;`	4	4	12
`b = temp;`	4	12	12

Topics for today (finish assignment 5)

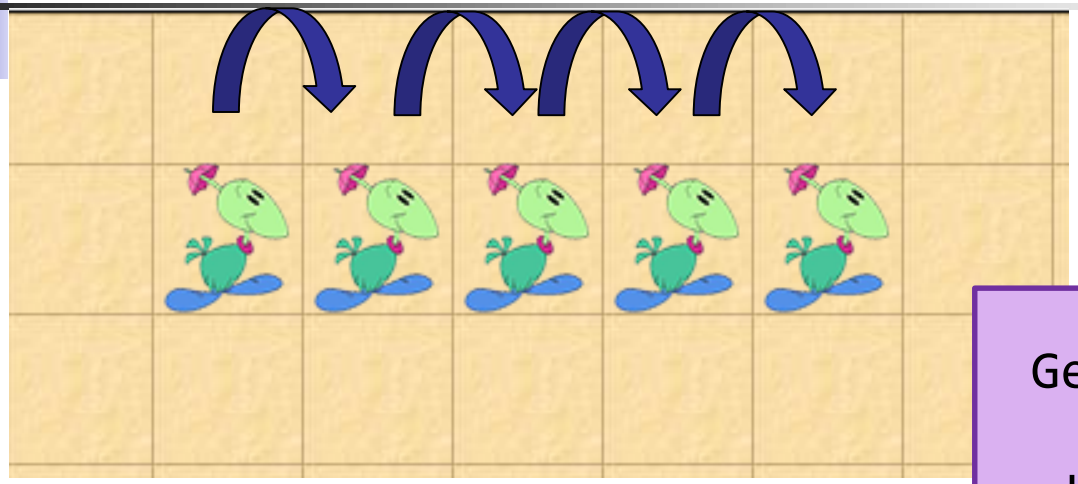
- Counter in while-loop
- Algorithms & efficiency

Variables and repetitions

- Use variables to remember things.
  - to repeat something several times
  - to remember how many times you already did it (or how times you still have to do it)

Repetition

Example: Mimi moves random times

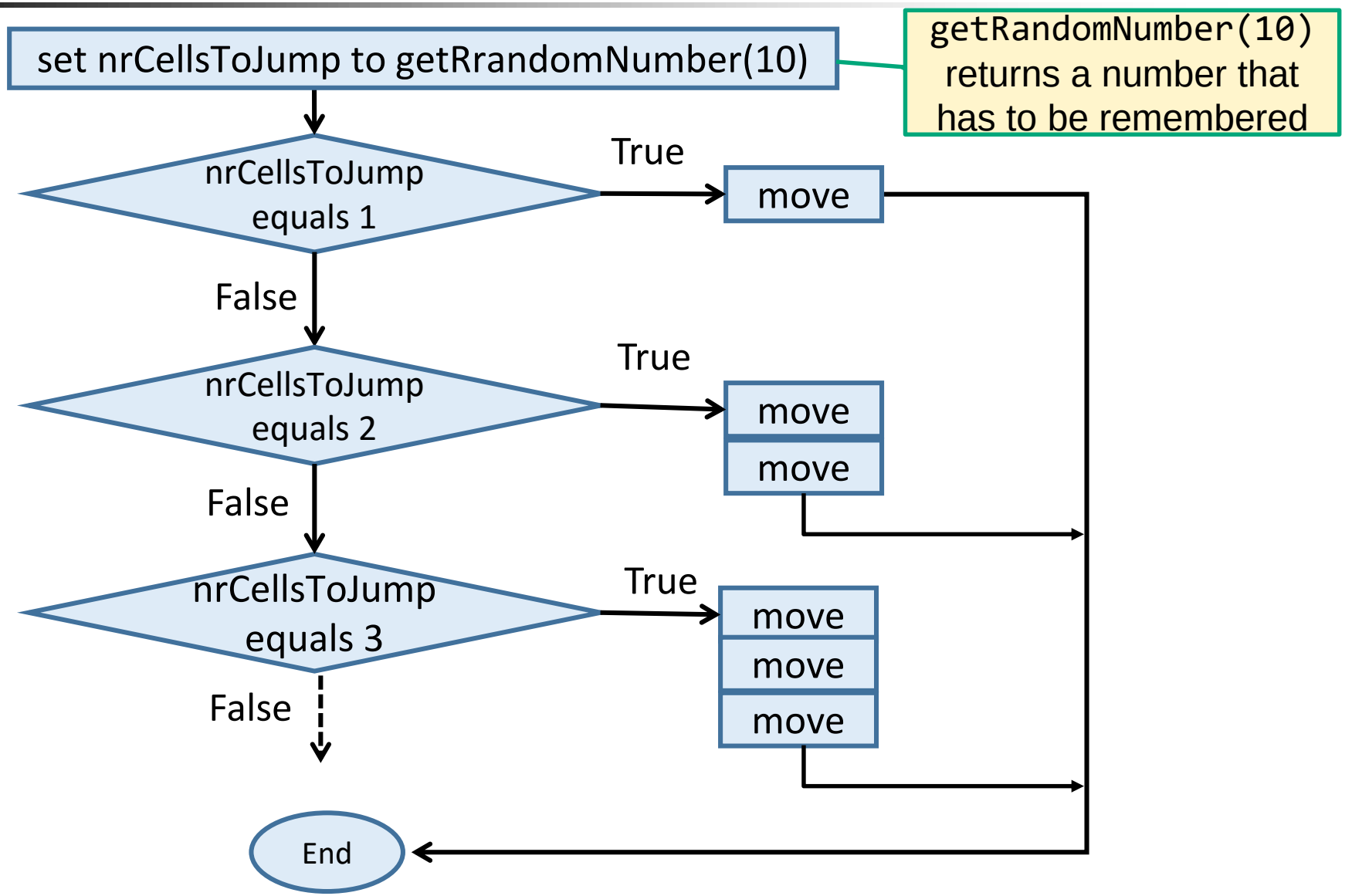


`GetRandomNumber(N)` will give a random number between `0` and `N` (`N` not included)

Sketch how would you make Mimi move forward a random number of 0-9 cells (`jumpRandomly` method) using:

- `getRandomNumber(10)`
- `a variable to remember how many moves must be made`
- `Dodo's move() method`

Nested `if … then … else` statements



Move a random number of times

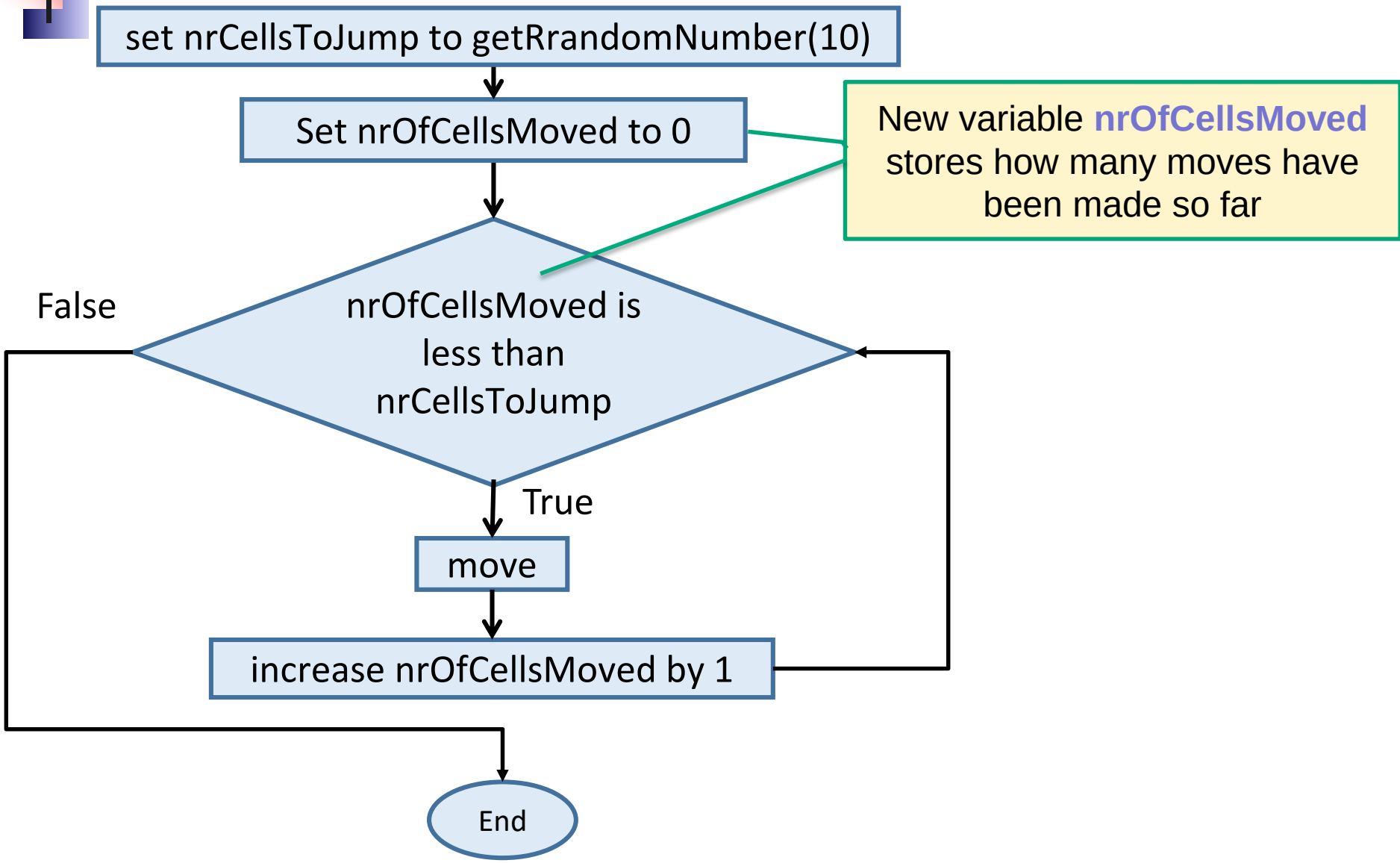
We use a (local) `int` variable with name `nrCellsToJump` to store the random number

`getRandomNumber(10)` returns a number that has to be remembered

```
public void jumpRandomly () {
   int nrCellsToJump = Greenfoot.getRandomNumber(10);
   if (nrCellsToJump == 1){
      move();
   } else if (nrCellsToJump == 2){
      move();
      move();
   } else if (nrCellsToJump == 3){
      move();
      move();
      move();
   }
   ...
}
```

Mind the difference:
= (assignment)
== (comparison)

… alternative with while and counter



… alternative with counter and while

```
public void jumpRandomly () {
   int nrCellsToJump = Greenfoot.getRandomNumber(10);
   int nrCellsMoved = 0;
   while ( nrCellsMoved < nrCellsToJump ){
      move ();
      nrCellsMoved = nrCellsMoved + 1;
   }

}
```

To store how many moves have been made so far.

The current value of `nrCellsMoved`...

... incremented and assigned to `nrCellsMoved`

Comparing with(out) counter & while

```
public void jumpRandomly () {
   int nrCellsToJump = Greenfoot.getRandomNumber(10);
   if (nrCellsToJump == 1){
      move();
   } else if (nrCellsToJump == 2){
      move();
      move();
   } else if (nrCellsToJump == 3){
      move();
      move();
      move();
   }
   ...
}
```

```
public void jumpRandomly () {
   int nrCellsToJump = Greenfoot.getRandomNumber(10);
   int nrCellsMoved = 0;
   while ( nrCellsMoved < nrCellsToJump ){
      move ();
      nrCellsMoved = nrCellsMoved + 1;
   }
}
```

Questions?

Quiz Next week

- Date: Feb 5th
- Topics:
  - Operators
  - Conditions
  - Return statements
  - Nesting
  - Decomposition/abstraction
  - Flowcharts
  - Variables
  - Tracing code

Wrapping up

Quiz on Feb 5th

Homework for Wednesday 8:30 Feb 3rd:

- Assignment 5:
  - **FINISH assignment 5**
  - ZIP code and ‘IN’ and **email** to **Renske.weeda@gmail.com**