

Sort the Cows

Setting the scene – a bit of the story



You are a farmer with a herd of cows and you need to organize those that are large enough to take to the market. You would like to select your 10 largest cows, or perhaps the smallest, the largest and the median, to get an idea of the overall size of the herd (an estimate of size like the average). You don't have a mechanism to weight cows, but you can put them through rails and sorting gates. In fact, you would like to build a sorting network with rails and gates to regularly examine the size of herds. How can you construct these rails so they use the least number of comparison gates?

The Activity

This activity is a competition between 4 or 5 teams of up to 5 participants each. The activity requires each team to create a sorting network of parallel rails (masking tape/ribbon/duct tape) and comparisons gates (different colour tape/ribbon) on a flat surface and test if several selection or sorting arrangements can be obtained no matter in what order the cows are lined up at the start of the network.

The activity has several phases:

1. The instructor gives a demonstration of what is expected. Distributes 4 long stripes of black cloth to simulate the rails. Distributes similar but shorter pieces of blue cloth and asks students to build a network that finds the smallest cow. Participants should be able to act as the cows in the network. They earn points by correctly selecting the smallest cow among 3 rails (then progress to 4 rails).



Sort the Cows

2. The teams experiment to create their own network. They earn points by illustrating their construction. These constructions are made with only the black rails and only the blue gates. These blue gates place the larger cow to the left. Specific problems to solve are
 - a. Find the largest cow (the largest cow is in a pre-determined rail no matter the initial order)
 - b. Find the smallest cow.
 - c. Find the smallest and the largest cow, not necessarily sorting.
 - d. Give the first problem solving set of exercises
 - i. Can gates that do not cross rails solve the above questions?
 - ii. What is a plan to check that the network is correct?
 - iii. If there are 5 rails, would it be possible to find the smallest cow with 3 rails, why yes or why no? In general, if there are n rails, is it possible to find the largest cow with $n-2$ or less gates?



3. Another variant is to ask team to swap and try to find an input where a network from another team fails and award points if such network is found.
4. Introduce participants to other types of gates. Green gates always place the smaller of the two cows on the left (or as per diagram above, the smaller cow goes up).
5. Provide the teams with the mats, black stripes and red, blue and green gates and ask similar questions and problems as above but now with networks of 8 rails. Allow participants to work on the floor or on the table to build larger networks.



Sort the Cows



6. Observe their constructions and provide information about the parameters of the problems (size of the network, cost of the network, varying cost of gates)
7. The instructor captures an image of each construction and uses the computer tool RailSort to
 - a. Capture the constructions by the students
 - b. Validate solutions and award them a score for different problems.
 - ▀ **Overall Network Score:** <A measure of the success of this network.>
---- SUCCESS ----
Runs: <The total number of inputs tested.>
Success: <The number of inputs that the network succeeded on, based on its current evaluation measure.>
Fails: <The number of inputs that the network failed on, based on its current evaluation measure.>

---- COST ----
Cost of this network: 10
- time: 5
- operations: 5
- parallel: 0
- bridges: 0
8. The instructor indicates how this activity translates to real world applications (Design of circuits) and IT (problem solving, algorithm design, teamwork, etc.)
9. If time permits, the instructor discusses the possibility of the sorting network becoming programs (with the API for this activity and the programming language **MaSH**). Alternatively, can illustrate building even larger sorting networks with RailSort.
10. Another variant is to limit the gates (red/blue green) and see if the solution can get the answer most of the time (for a large number of permutation).

Requirements

1. Instructor and assistant
2. Up to 20 participants
3. Long black cloth to represent rails. Blue cloth pieces to represent gates.
4. Masking tape / back ribbon (rails), green, blue and red ribbon for gates, white background to capture participants' constructions.
5. Safety scissors – 5 (1 per team)
6. 8 cow markers per team (all of different size) Notepads – 5 (1 per team)
7. Pens – 5 (1 per team)
8. Whiteboard markers – 3 (Black, blue, red) for instructor
9. This instruction sheet –
10. Instructions for participants (initial)---7 (1 per team, 1 for each instructor)

Sort the Cows

11. Questions for participants – 7 (1 per team, 1 for each instructor)
12. Feedback form.

Time Frame (90 -120 minutes)

The expected time frame for each of the phases is:

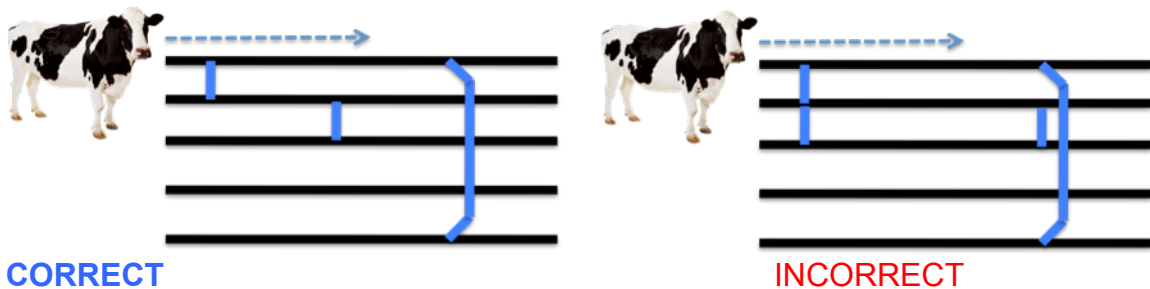
1. Instructor demonstration and distributing materials (10 minutes)
2. a. Making the network to find largest, then largest and smallest with 3 rails, participants walk on it (15 minutes), and extension to 4 rails, 10 minutes. Writing solutions to first phase of problems (10 minutes).
3. a. Distributing ribbon and background, explaining problems and new gates, problems must be solved for at least 8 rails, but participants can start with 3 or 4 rails and progress from there (5-10 minutes)
 - b. A network that finds the largest cow and places it in the middle rail. (10 minutes).
 - c. A network that places the smallest besides the largest at the left-most two rails (10 minutes).
 - d. A network that finds the median --- has the same number of larger than smaller for an even number of rails (10 minutes)
 - e. A network that sort from smallest to largest.
4. Instructor captures an image of each team's network and illustrates to each team the scores (3 to 4 minutes each)
5. Instructor discussing practical aspects of the activity (5 minutes).

Phase 1 – Demonstration

The instructor will provide a brief demonstration using the following sample roads and intersections:

Rules for creating the network (3 or 4 rails walking on it):

- Each backcloth represents a rail where only one cow can travel. The rails are laid out parallel to each other. Gates are laid out orthogonally.
- Two gates cannot connect in the same position for the same rail
- A gate can connect across rails that are not adjacent, but it may cost more.



Rules for creating a network on a mat with more than 4 rails:

- Same as for those where people work on, but now there are also red gates and green gates.

Phase 2 – Physical experience

You may need to guide the participants and assist them to use the material given (some notes on problem solving and some questions)

Sort the Cows

Phase 3 – Construction with ribbon on a mat (above 4 rails)

You may also need to assist with the problems. See documentation of RailSort.

Phase 4 – Relating this activity to the real world

1. The instructor will outline a successful strategy to meet the objective.
2. The instructor will indicate how the activity relates to real world problems. It can introduce the idea of *algorithm*. The instructor may discuss notions for what is a problem for IT and how an algorithm solves it for all instances. May discuss the points in the participants' summary regarding problem solving. It also can relate to ideas of cost of computation and cost of operations (notion of time for an algorithms, or parallelism). If the complementary activity, initial concepts of programming in textual from can be introduced. Thus new concepts could be:

Concepts:

- identifier
- method invocation
- sequential control
- comments
- abstract execution
- displaying output

With level contro,;

For-loop

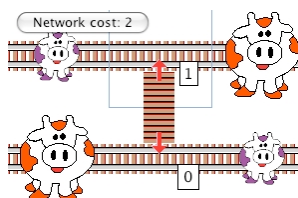
With **MaSH** and the virtual tool

- constants
- arrays
- sorting algorithms
- complexity

OPTIONAL PHASE – Construct a description of your city with textual programming language.

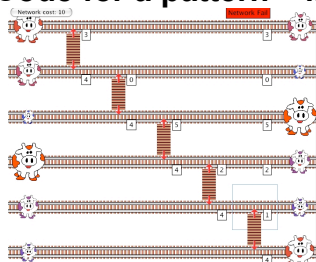
See documentation of MaSH sortingNetwork environment.

Code for one gate – MaSH level statements



```
import sortingNetwork;
createRails(2); // create two rails with data in random order
printOrder(); // print the data
redSwap(0,1); // Swap values in position 0 and 1
printOrder(); // print the data
```

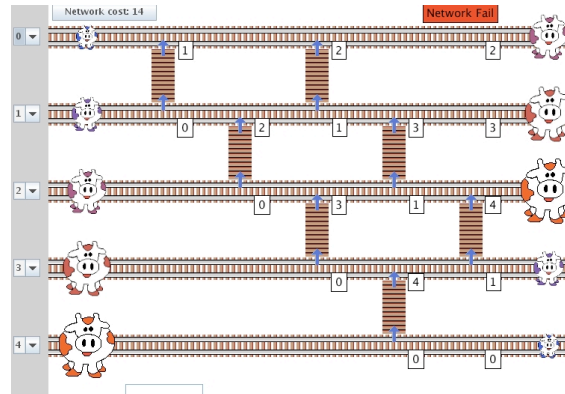
Code for a pattern – Mash level control



```
import sortingNetwork;
createRails(5); // create 5 rails with data in random order
printOrder(); // print the data
// cyclically shift one position to the left
for (int i=0; i<4; i=i+1)
    redSwap(i,i+1);
printOrder(); // print the data
```

Sort the Cows

Repeat a Pattern



```
import sortingNetwork;

void select (int place) {
    for (int i=0; i<place; i=i+1)
        blueLargerUp(i,i+1);
}

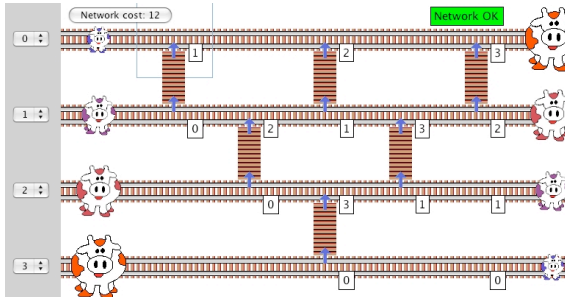
void main() {
    createRails(5); // create five rails with data in random order

    printOrder(); // print the data

    // Select and place smallest at the bottom
    select(4);
    select(3);
    select(2);
    select(1);

    printOrder(); // print the data
}
```

MASH level methods:



```
import sortingNetwork;

void select (int place) {
    for (int i=0; i<place; i=i+1)
        blueLargerUp(i,i+1);
}

void sort (int place) {
    for (int i=place; i>0; i=i-1)
        select(place);
}

void main() {
    createRails(5); // create five rails with data in random order

    printOrder(); // print the data

    sort(4); //Sort

    printOrder(); // print the data
}
```