

Sample Solutions

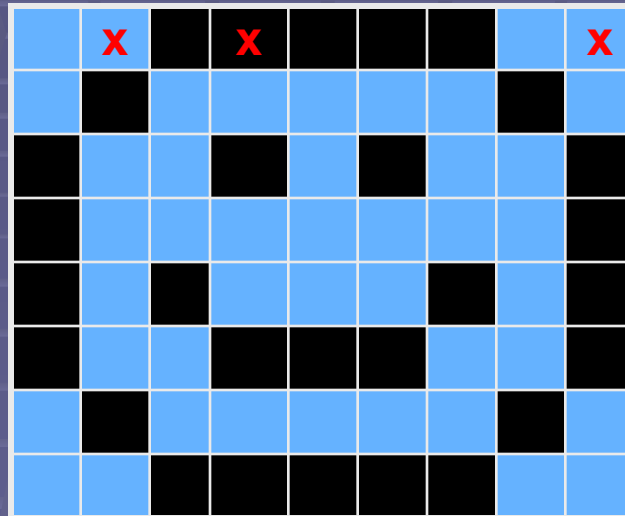
Central Europe Regional Contest 2007

Billboard – Problem



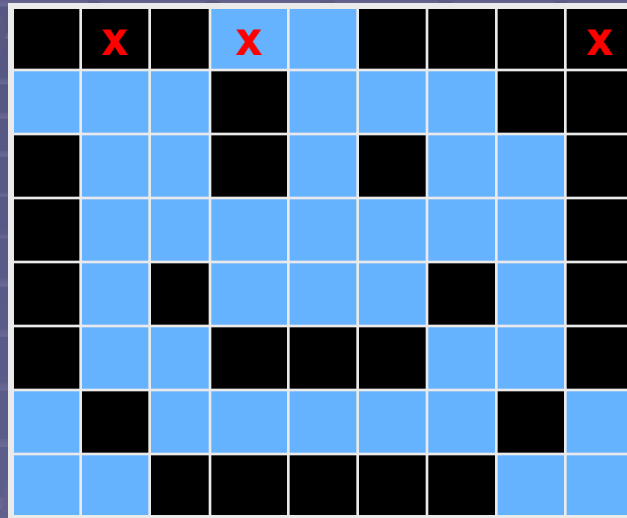
- Test all existing subsets
- 2^{256} possibilities for the 16x16 board
- 😞

Billboard – Main Idea



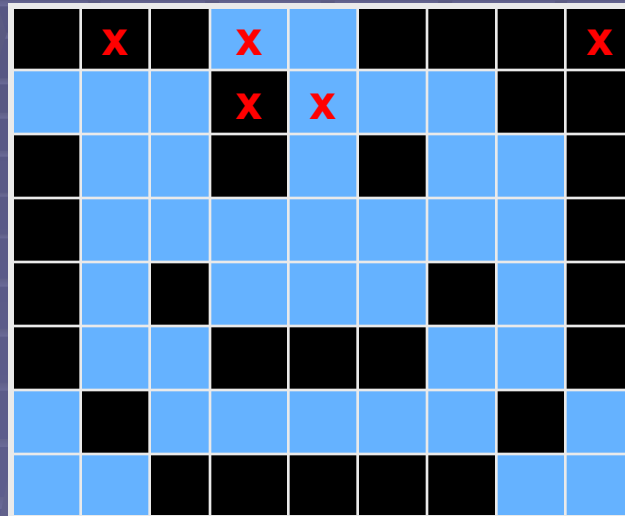
- What if we somehow knew the tapped tiles in the first row?

Billboard – Main Idea



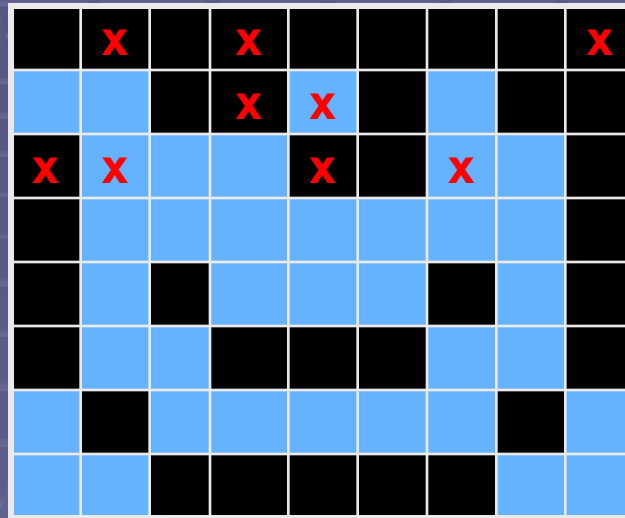
- We can try what happens with the row
- **WOW!** It is easy to find the second one!

Billboard – Main Idea



- We can try what happens with the row
- **WOW!** It is easy to find the second one!

Billboard – Main Idea

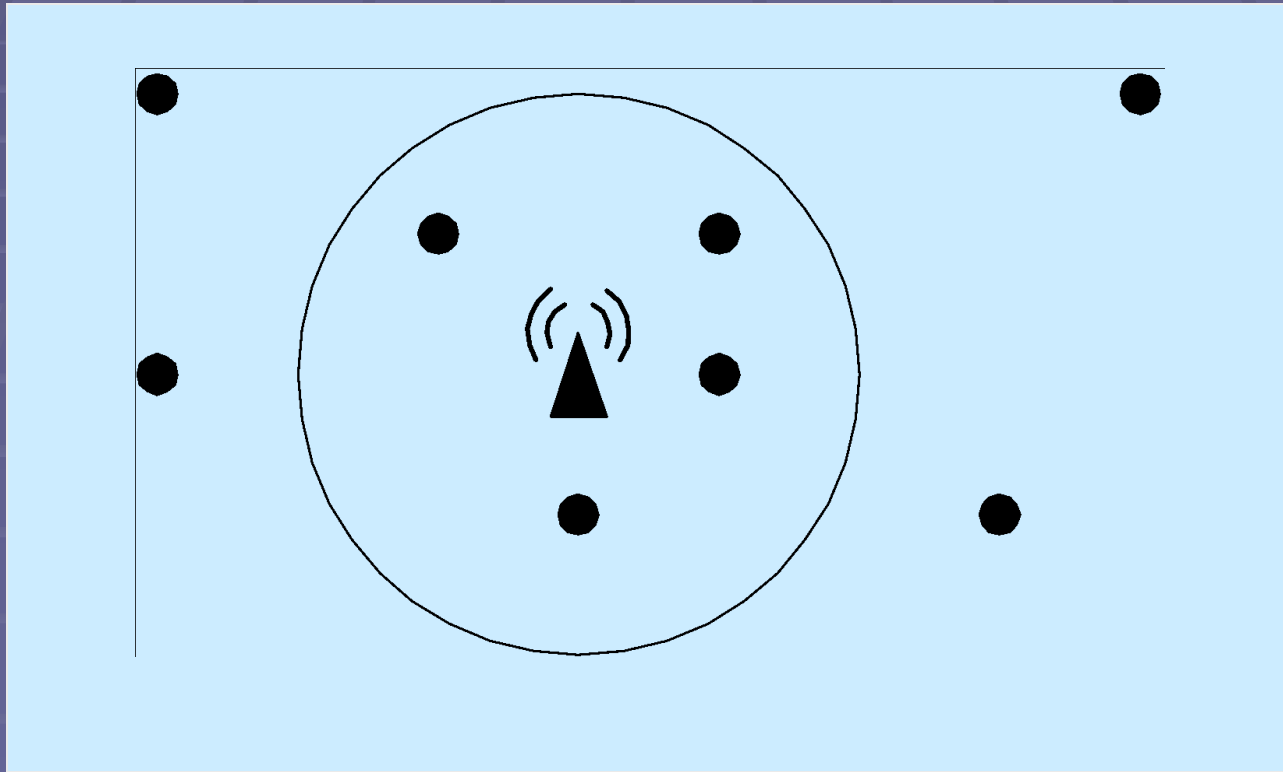


- Let's also do it...
- The third row becomes evident
- ... and so on

Billboard – Solution

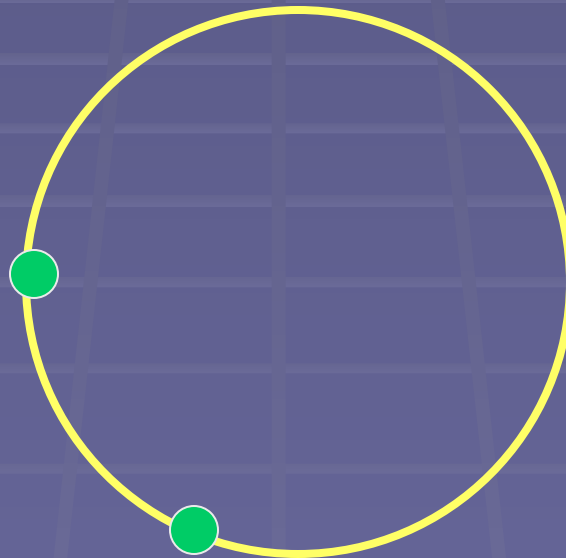
- Ok, but how to find the first row?
- We will try all possibilities!
- Approx. $2^{16} * 16^2$

C: Phone Cell



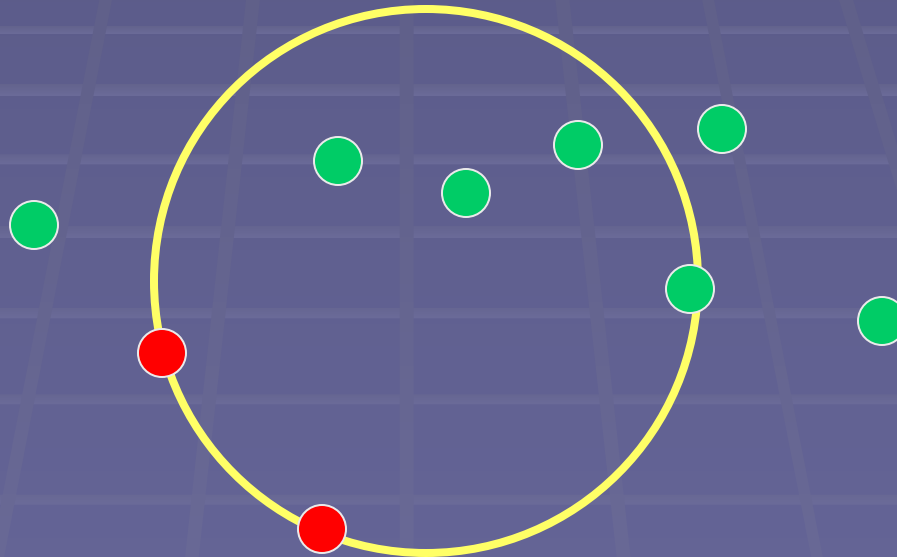
Cell – Problem

- Find a circle that covers most points
- Center may not be in an existing point
- □ Solution will always touch 2 points



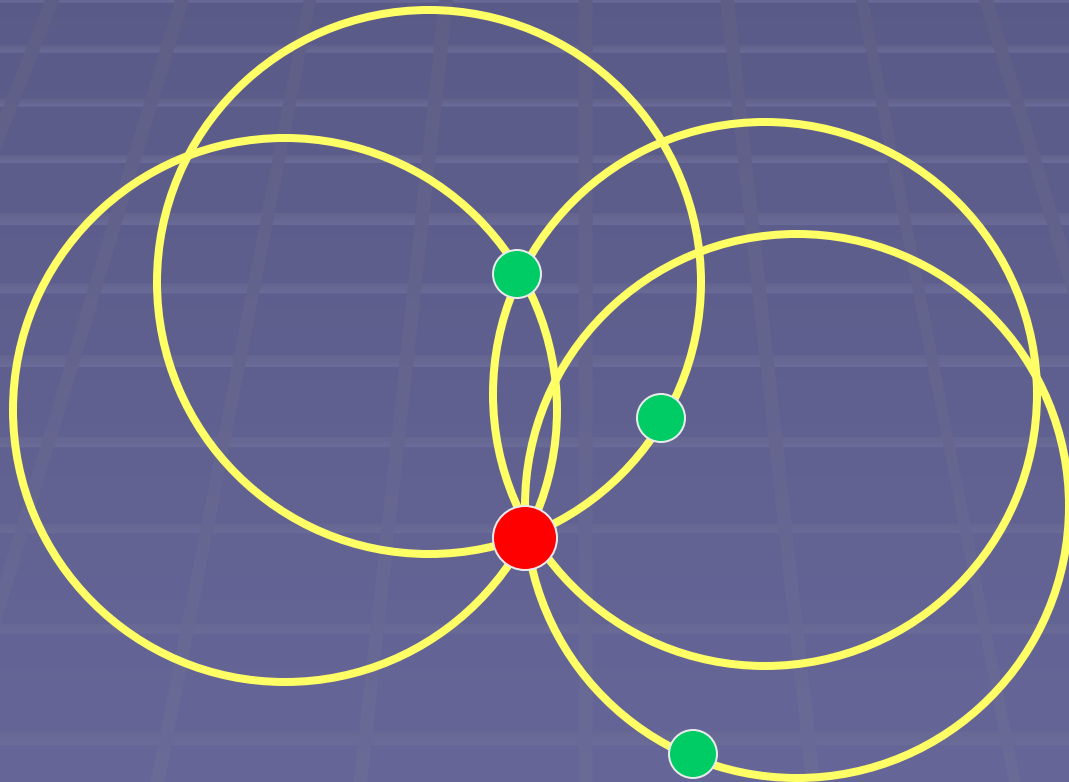
Cell – Simple Solution

- Take all pairs, find a circle
- Then all other points must be tested
- □ Time: n^3 😞



Cell –Solution

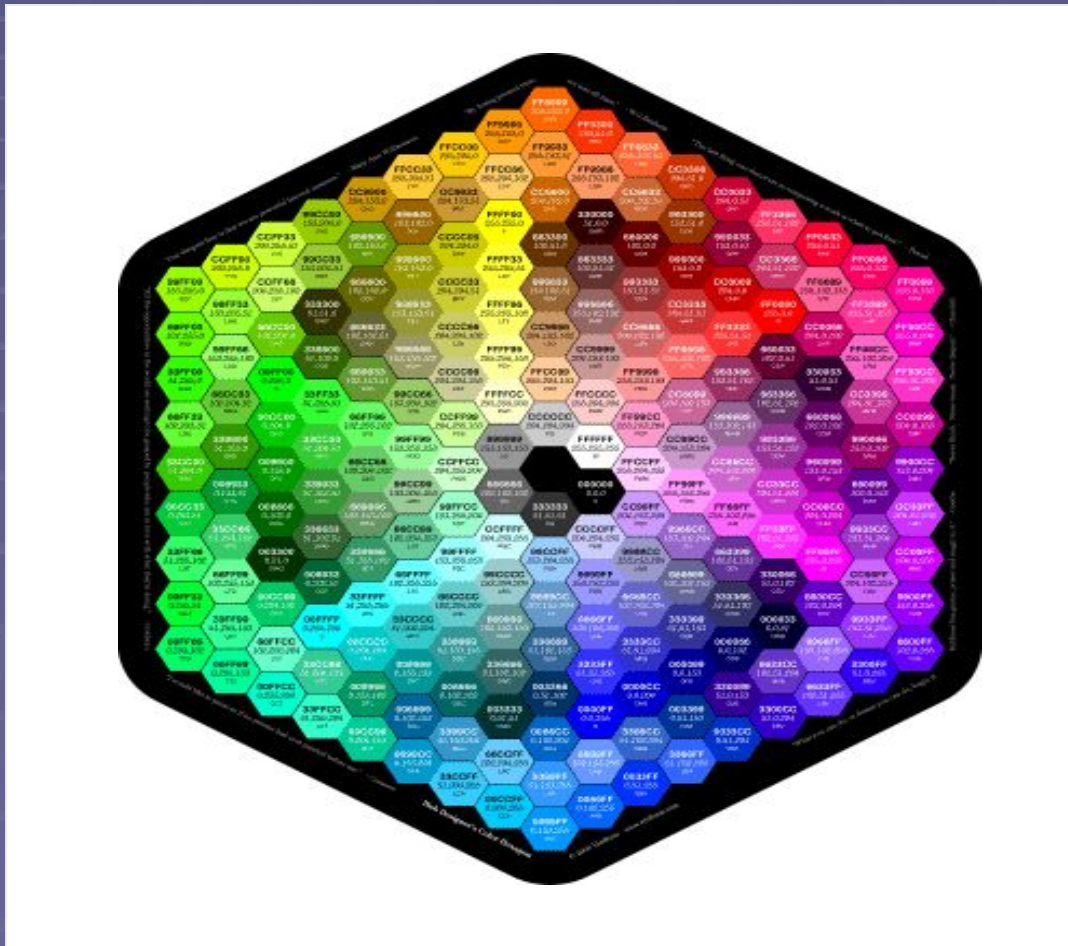
- For each point, use “**sweep**” technique
- Find „interesting angles“ and sort them



Cell –Solution

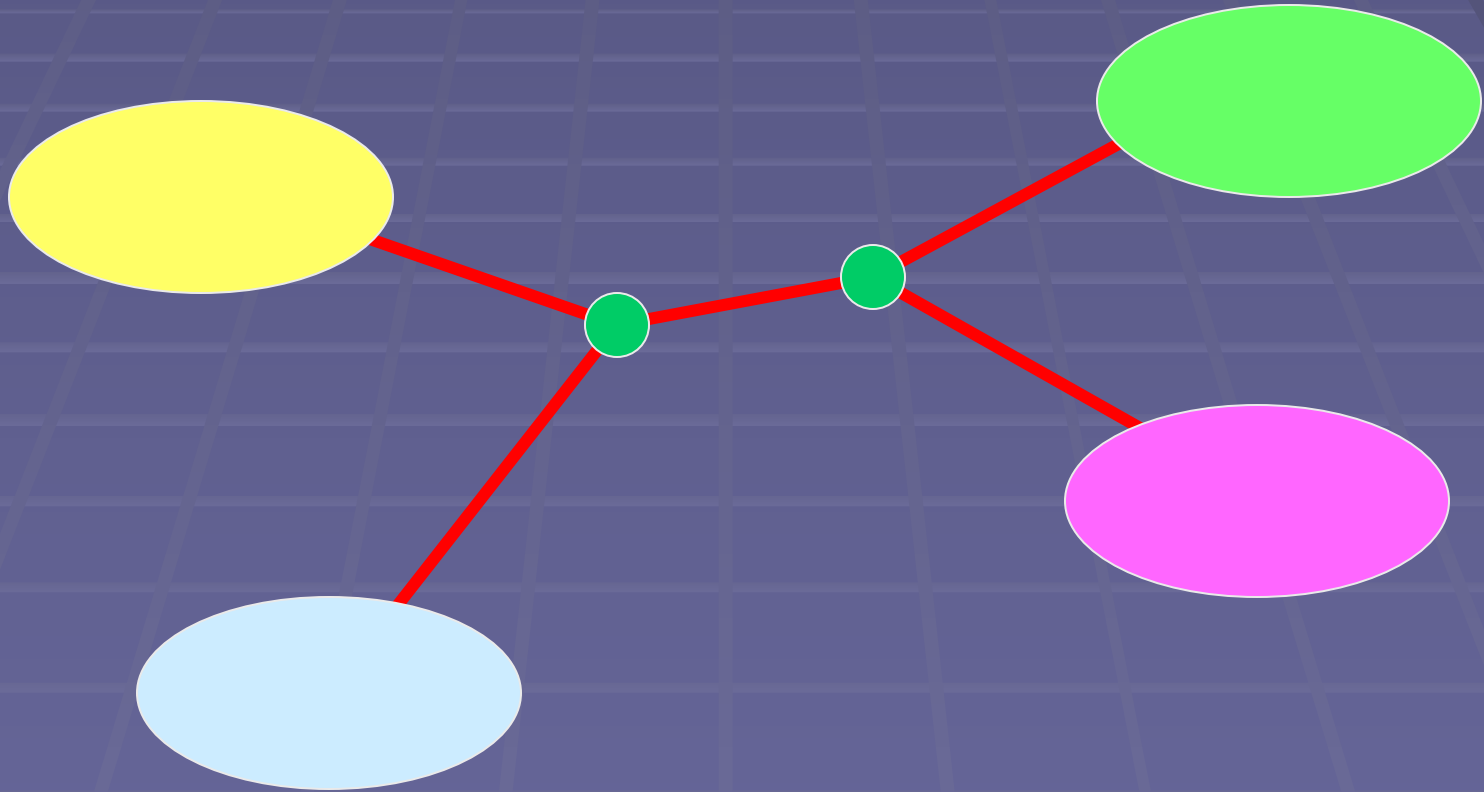
- One sorting for each point
- □ Time: $n^2 \cdot \log n$
- Carefully with floating point numbers!

H: Hexagon



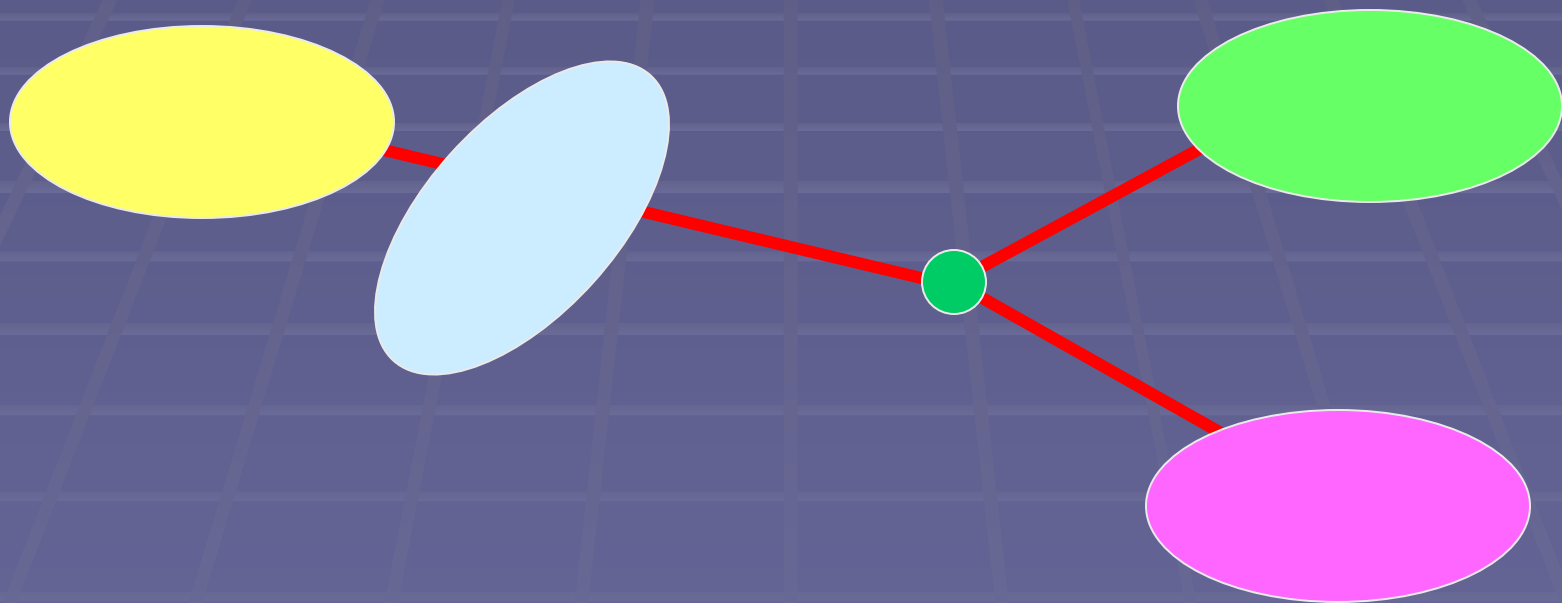
Hexagon – Idea

- All combinations of 2 green points



Hexagon – Solution

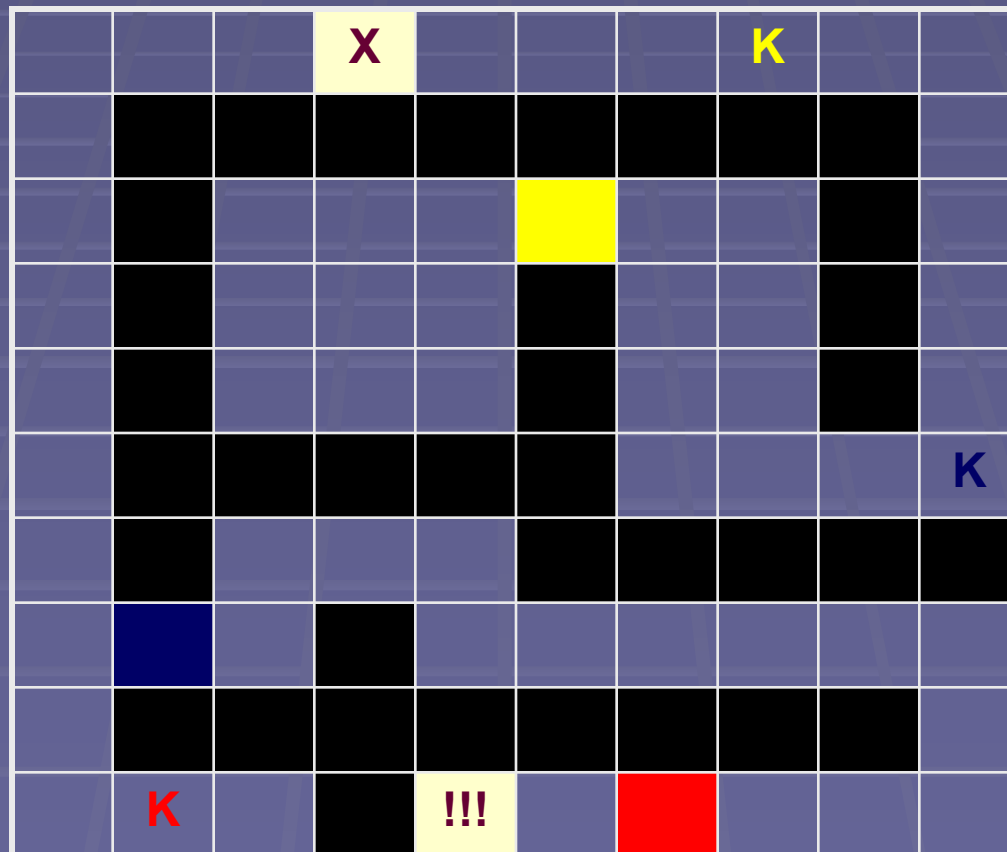
- Handle “special” cases



Hexagon –Solution II

- **Dynamic Programming**
- **For all area subsets (16), find the best solution using each parcel**

K: Keys

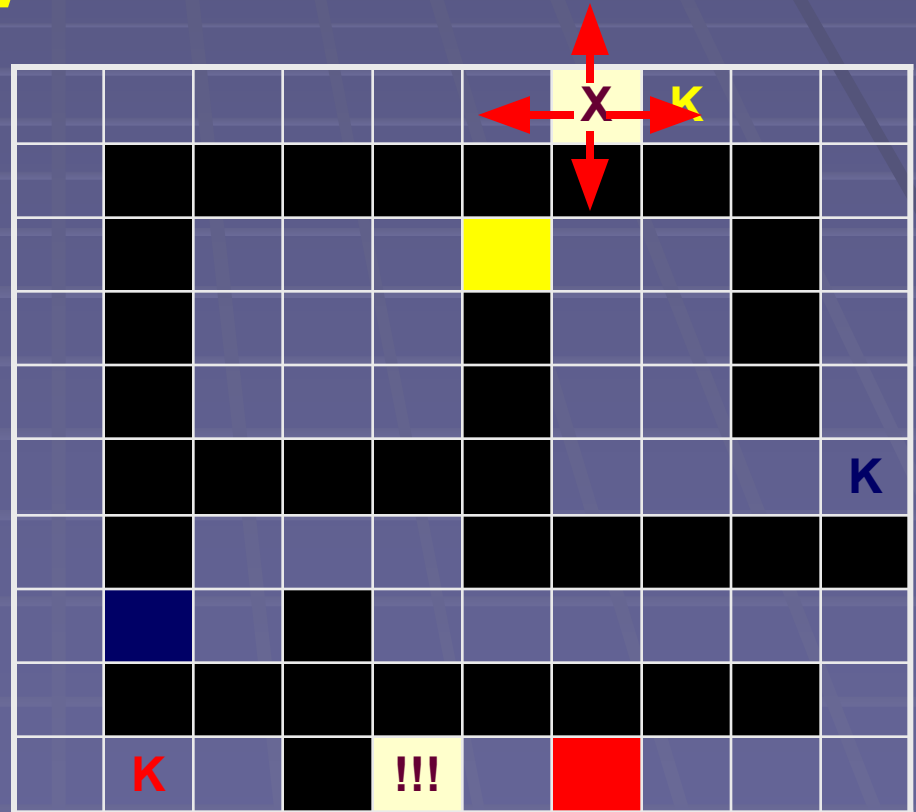


Keys – Idea

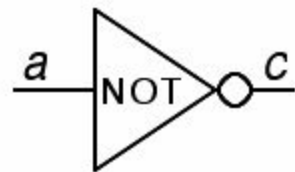
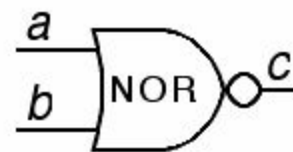
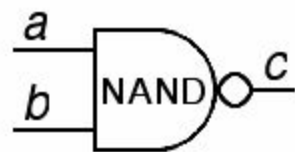
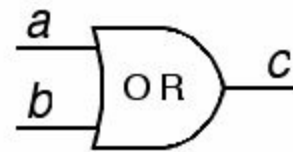
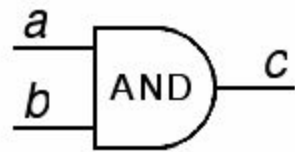
- **Breadth-First search**
- **Combination of position and keys (!)**
- **4 keys => 16 combinations**

Keys – Idea

- Position (1,7) + **Red** Key
 - (1,8) + **Red/Yellow**
 - (1,6) + **Red**



L: Logic



Logic – Problem

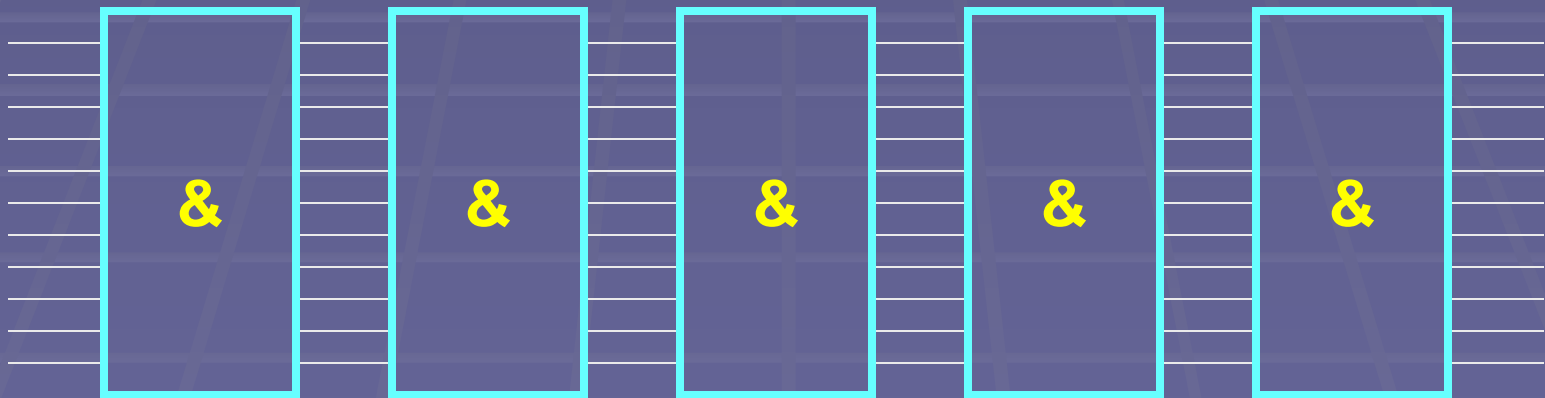
- **Problem? There is no problem**
- **“Only” follow the connections**
- **Compute logical operations**

Logic – Potential Pitfalls

- Gates with no input
 - AND □ 1
 - OR □ 0
 - XOR □ 0

Logic – Potential Pitfalls

- Splitting and joining paths
- => We must remember, what has already been computed



N: Numbers

$$10101010100101_{-2} = -10907$$

Numbers – Solution

- **TO** decimal:
 - Use the formula in the problem statement

$$4257_{-10} = -4000 + 200 - 50 + 7 = -3843$$

Numbers – Solution

- **FROM** decimal:
 - Use remainders (modulo)
 - Careful with the negative numbers!

- $4237 \bmod 10 = 7$ $\rightarrow 423$

- $-423 \bmod 10 = 7$ $\rightarrow -43$

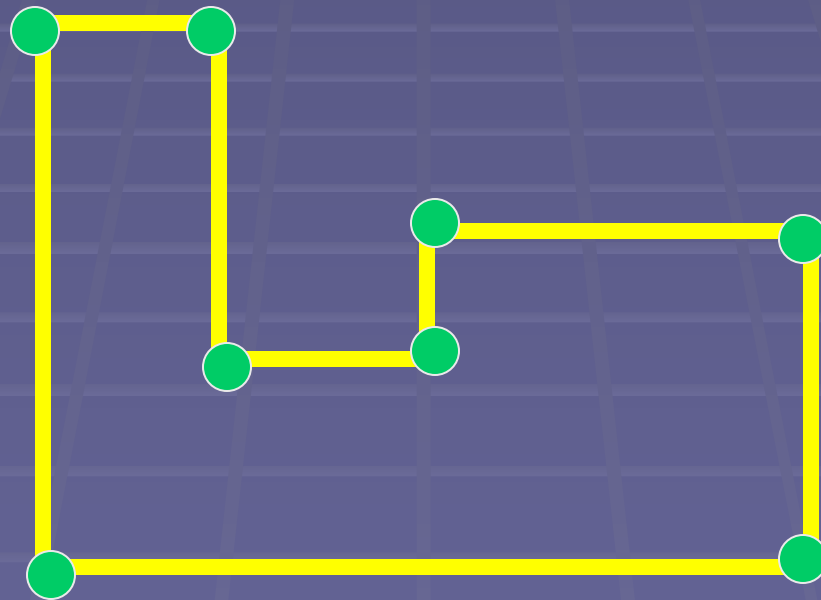
- $43 \bmod 10 = 3$ $\rightarrow 4$

- $-4 \bmod 10 = 6$ $\rightarrow -1$

- $1 \bmod 10 = 1$

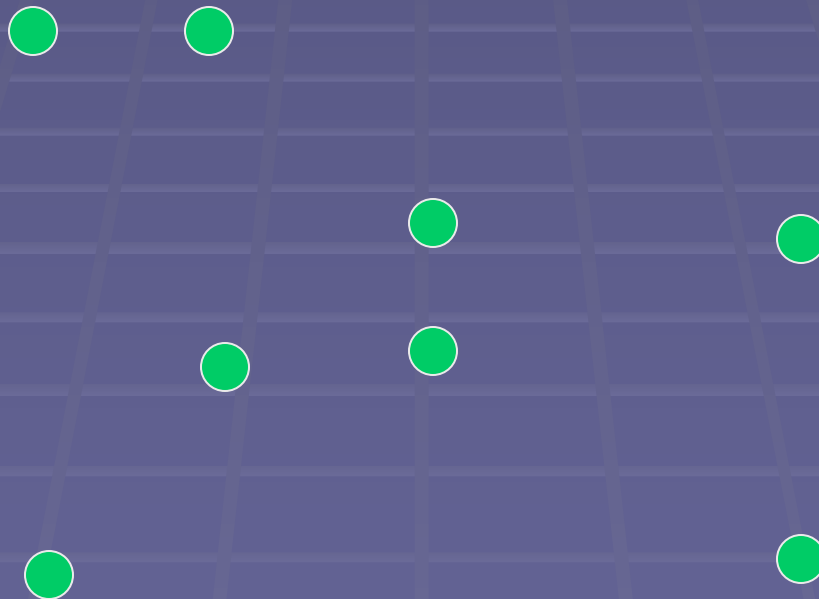
$\rightarrow 16377$

P: Polygon



Polygon – Problem

- Find the order of polygon vertices



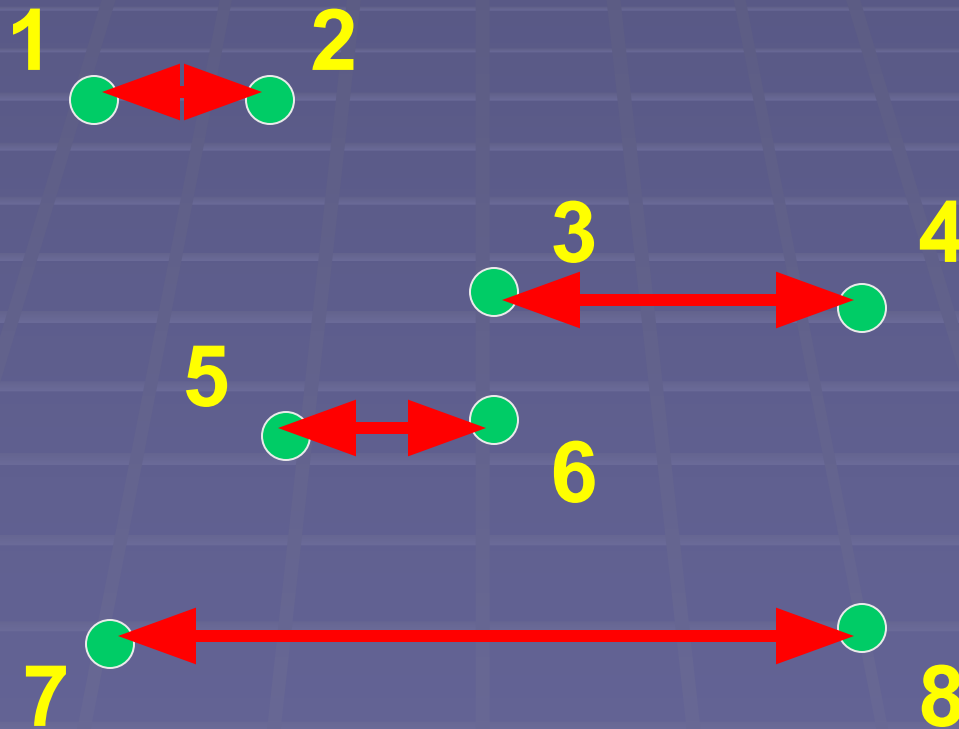
Polygon – Idea

- Sort the vertices by their X coordinate



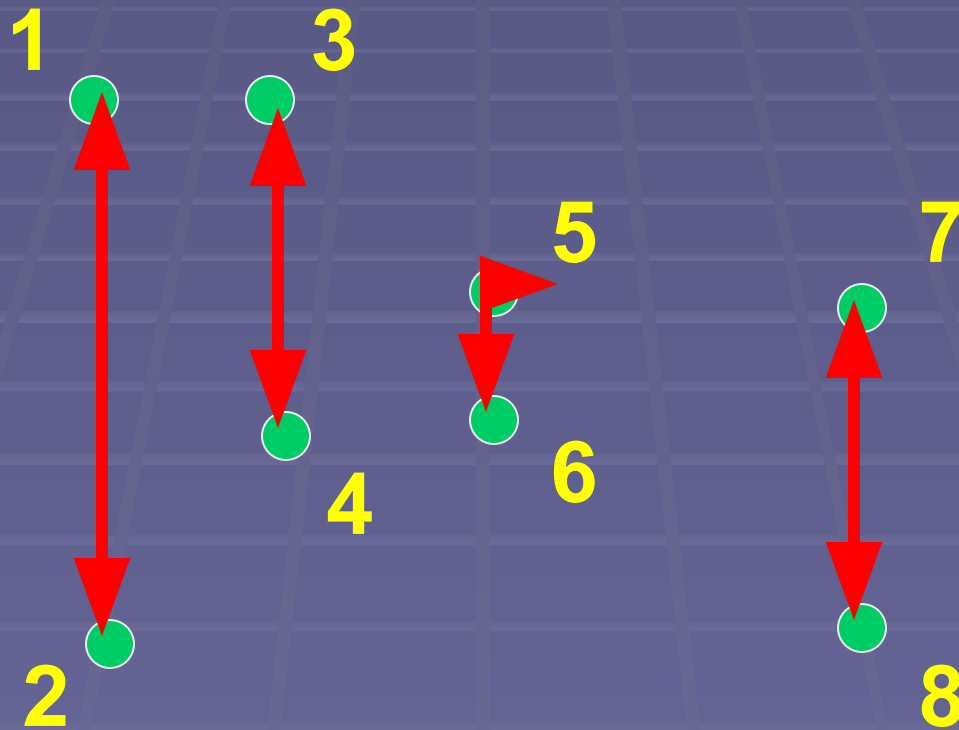
Polygon – Idea

- => Find their horizontal neighbor



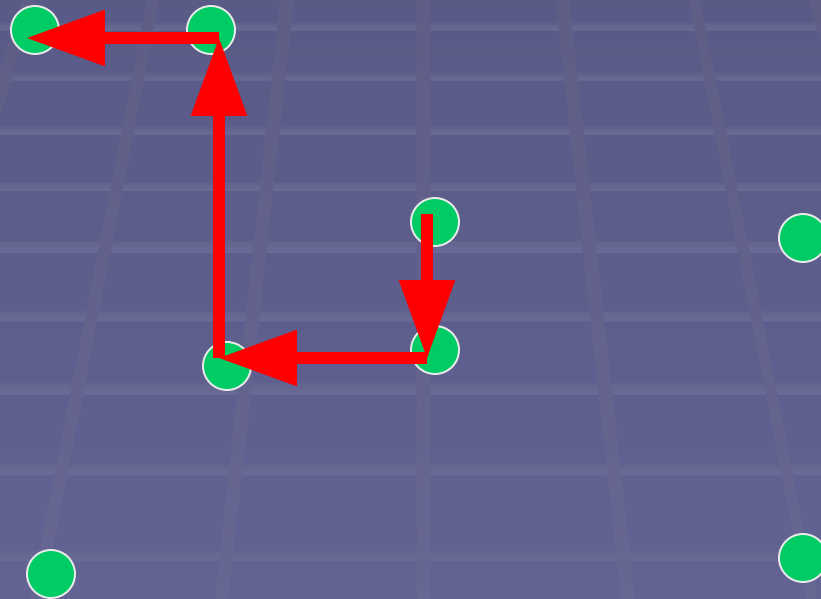
Polygon – Idea

- Sort by Y => vertical neighbor



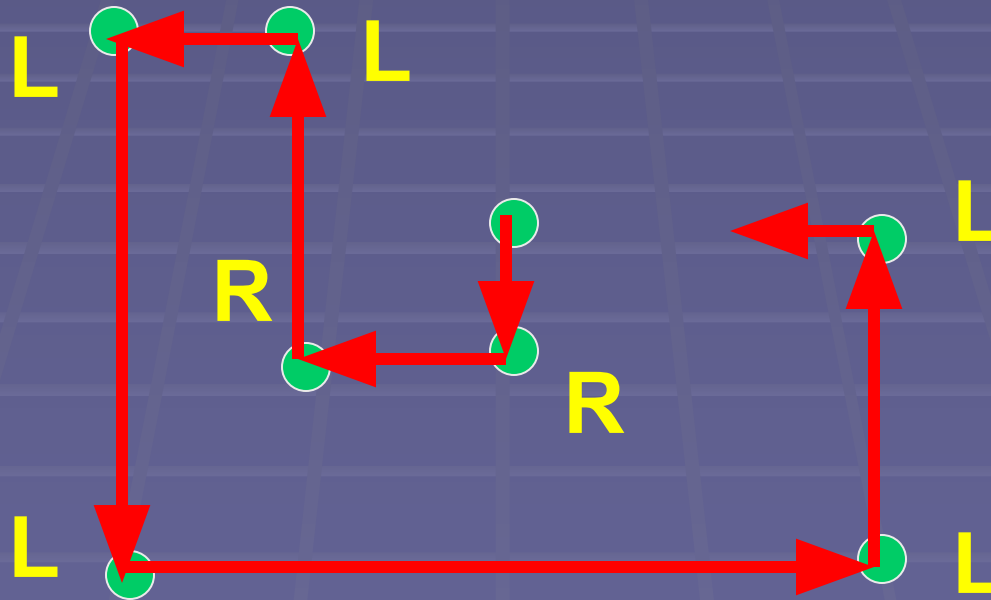
Polygon – Solution

- Now, each vertex knows its neighbors
- Start with the first, and walk around

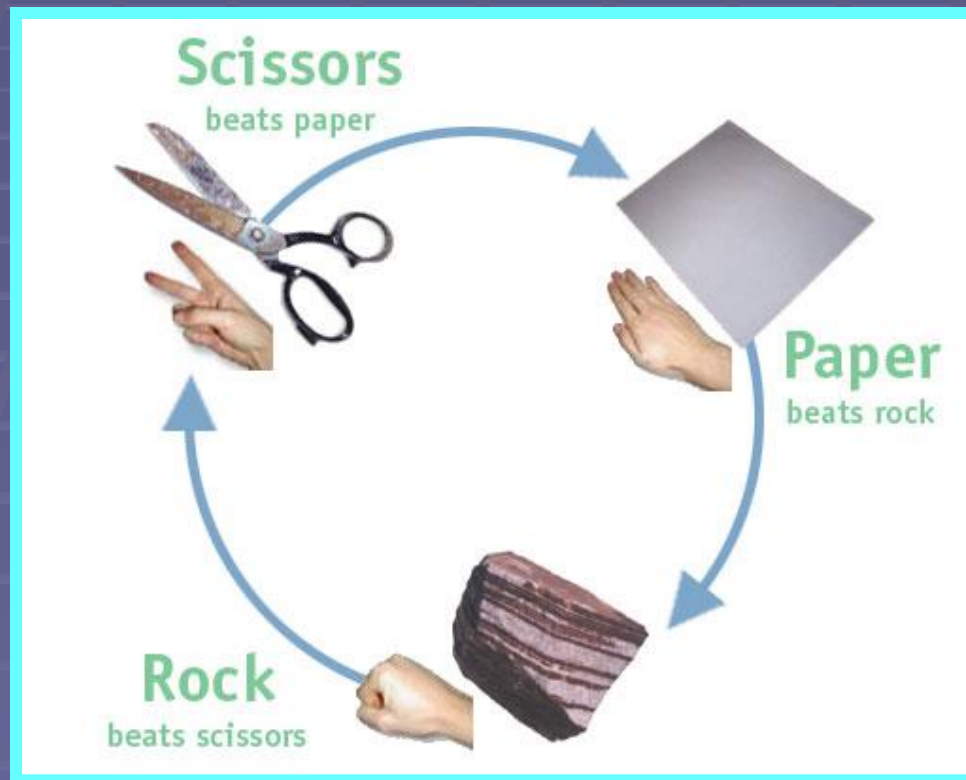


Polygon – Solution

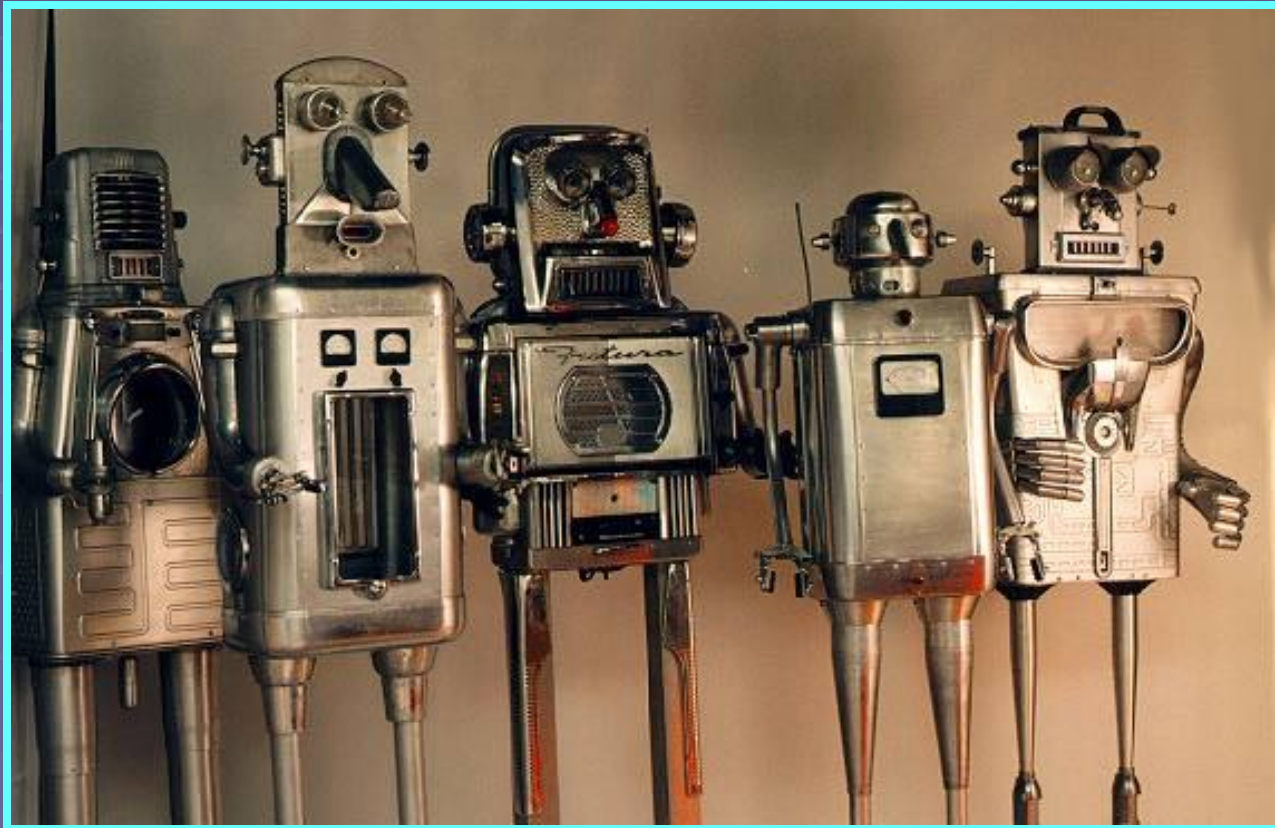
- Count Left / Right turns
- Left > Right ? => counter-clockwise



R: Roshambo



S: Robotic Sort



Sort – Problem

- Naive approach – reverse in an array
- Little bit better: remember reversed
- Quadratic time $\square 100\,000^2$ 😞

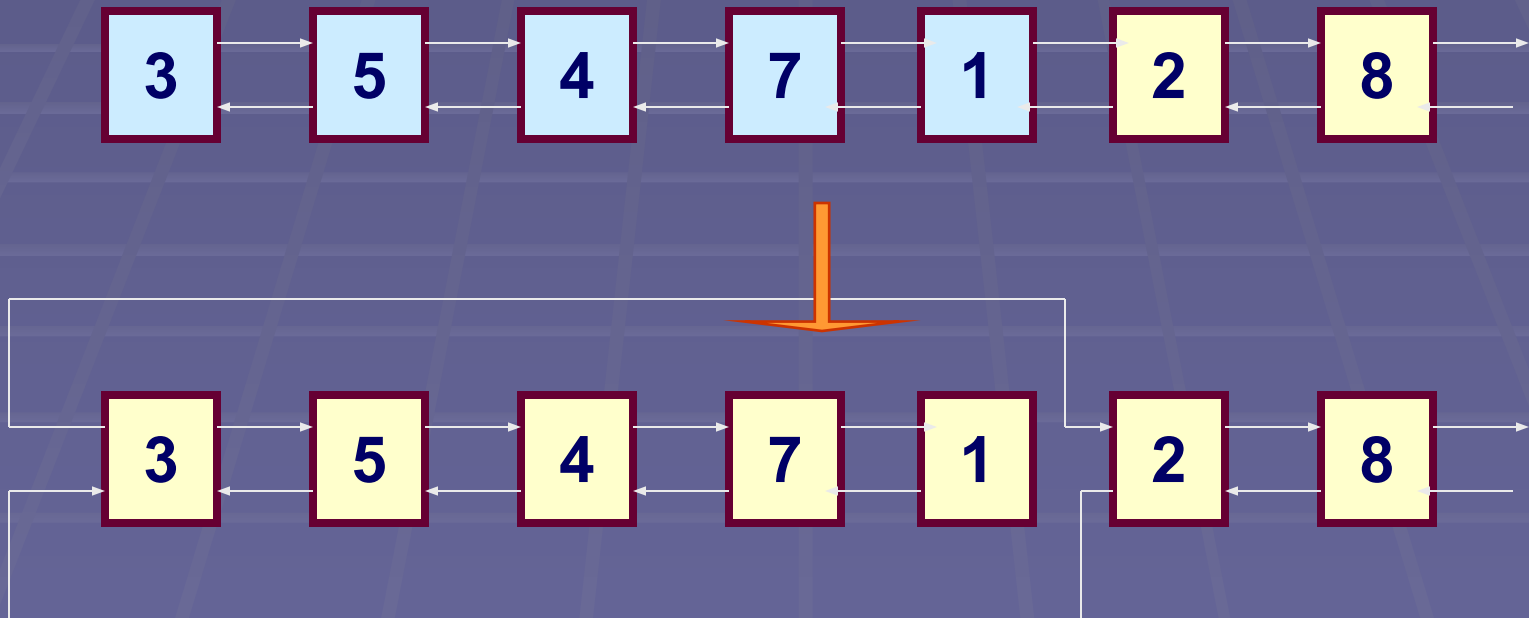
3	5	4	7	1	2	8	6
---	---	---	---	---	---	---	---



1	7	4	5	3	2	8	6
---	---	---	---	---	---	---	---

Sort – Another Approach

- Double-linked list
- Problem finding the „forward direction“



Sort – Solution

- Combined solution: use both
- Linked list + array of reversed
- The array time: $O(\text{rev_before})$
- After $\text{SQRT}(n)$ steps: reorder
- □ $n \cdot \text{SQRT}(n)$ time

Sort – Solution II

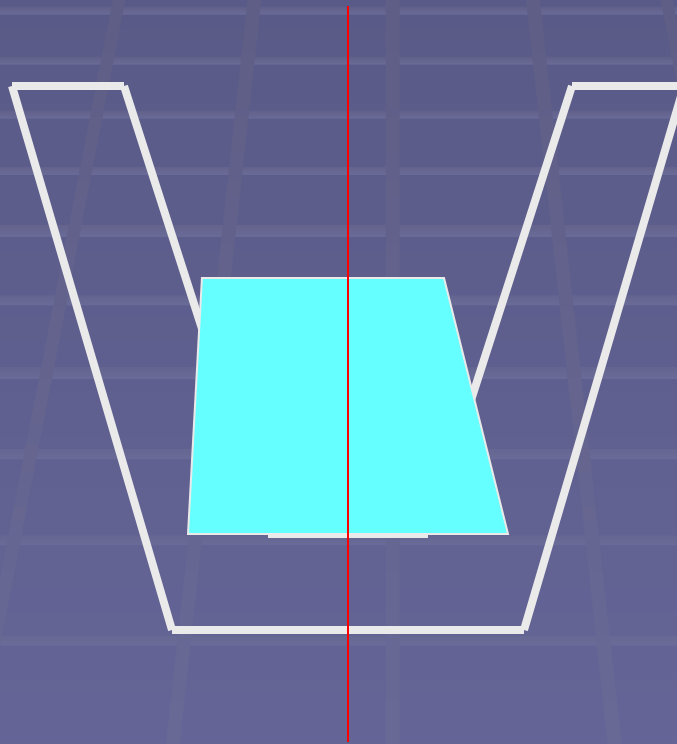
- **Other possibilities - ???**
 - **Heap**
 - **Interval Trees**
 - **... ?**

W: Water



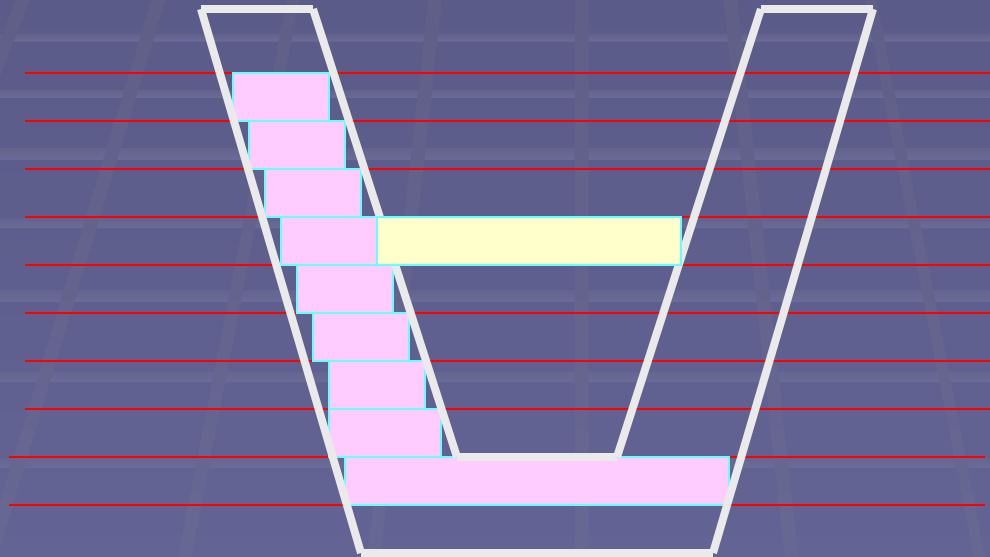
Water – Problem

- Find the **Center of Mass**
- Amount of water with the lowest center



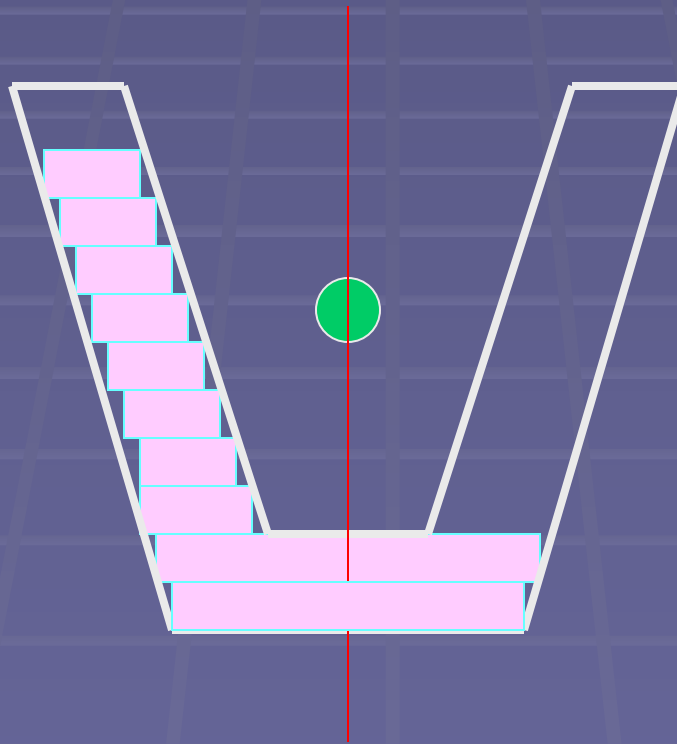
Water – Idea

- Split the glass into horizontal slices
- Volume of one slice: $\pi.r^2.d$



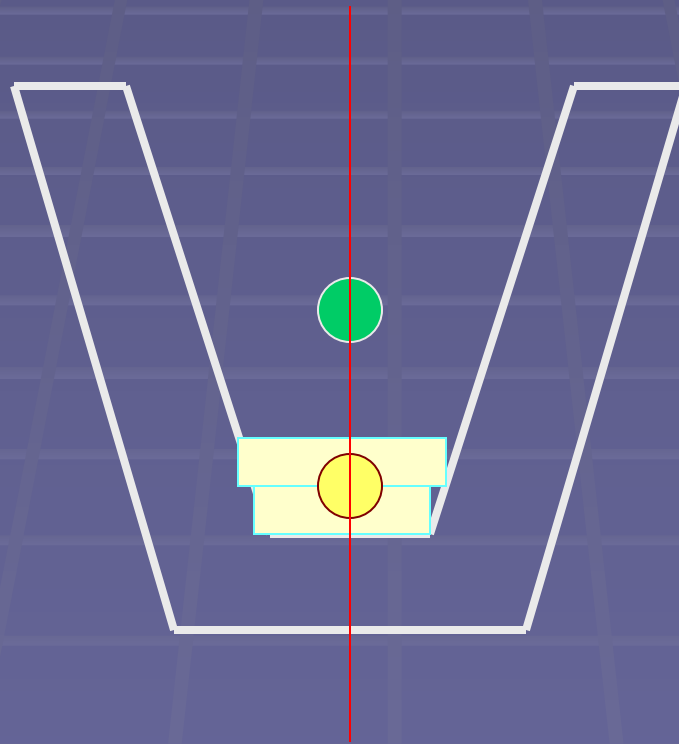
Water – Solution

- Center of Mass of the glass
- Weighted average of all slices



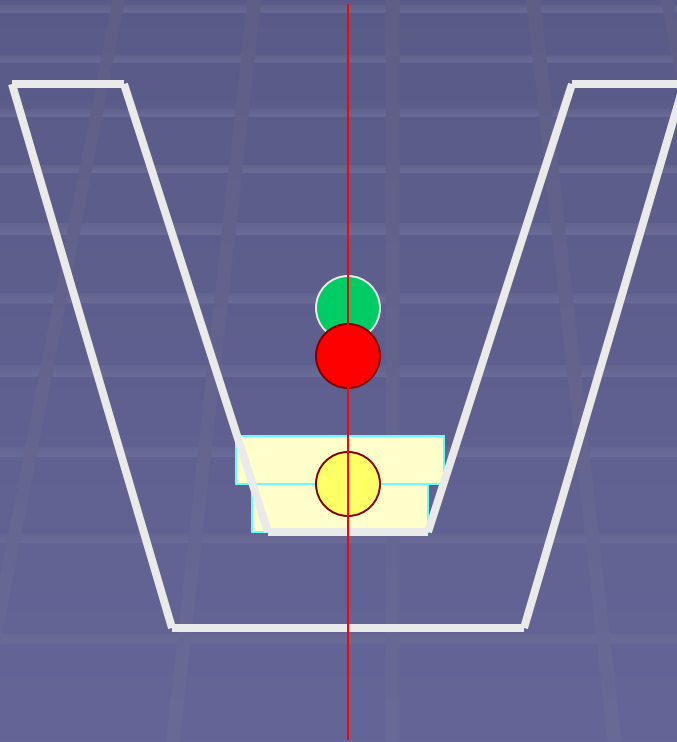
Water – Solution

- Add the water “slowly”
- Center of water mass



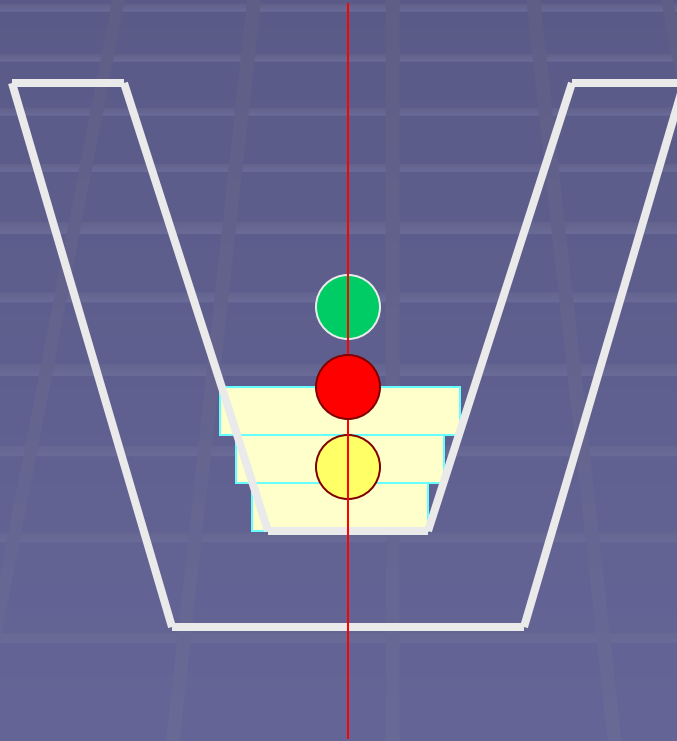
Water – Solution

- Combine both centers (weighted average)



Water – Solution

- Water reached the center ? => **END!**
- (Linear time needed only)



Statistics

	RT	TL	WA	PE	AC
Billboard	1	11	8	1	22
Cell	5	17	17	1	12
Hexagon	4	7	13	0	4
Keys	1	10	14	3	42
Logic	1	1	31	0	4
Numbers	11	5	65	0	46
Polygon	6	11	24	0	37
Roshambo	6	1	10	16	66
Sort	3	24	0	0	1
Water	1	2	1	0	1