Sample Solutions

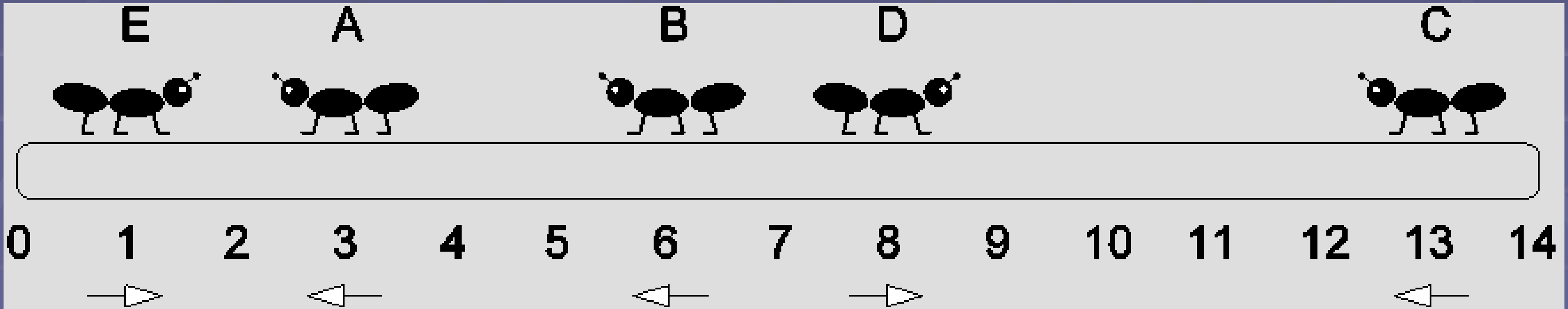
acm International Collegiate Programming Contest

IBM event sponsor

CENTRAL EUROPE REGIONAL CONTEST 2011

Czech Technical University in Prague

PRACTICE: ANTS



Ants

- Ants are “interchangeable”

- => Meeting and “turning around” can be ignored

- => Solution is trivial

PRACTICE: ELECTRICIAN

Electrician

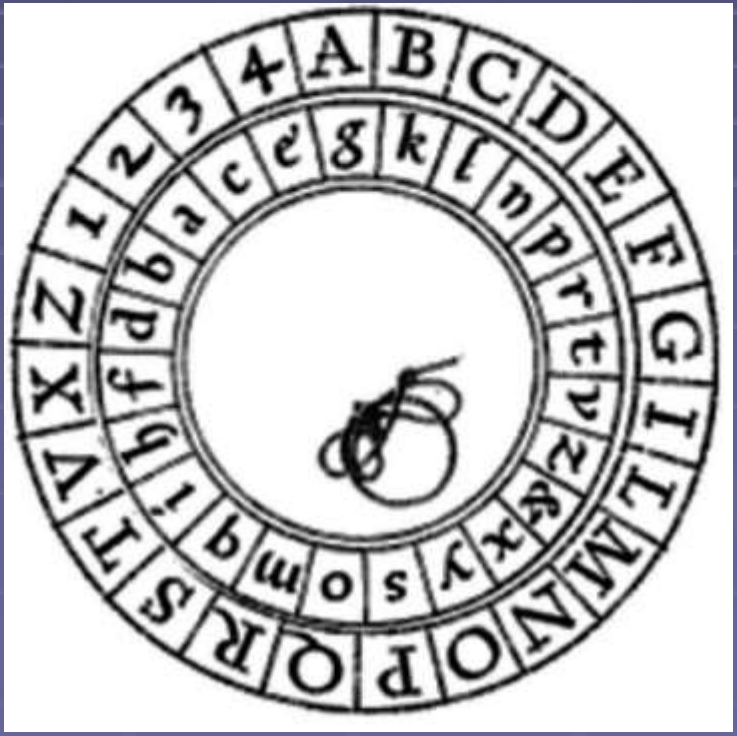
```
for (;;)
{
  scanf("%d", &x);
  if (x == 0) break;
  printf((x == 2)
     ? "Bad luck!\n“
     : "Electrician needs 1 trips.\n");
}
```

Sample Solutions

- Cards
- Vigenere
- Unique
- Trail

- Program
- Regulate
- Analyse
- Grille
- Unchange
- Execute

VIGENERE GRILLE

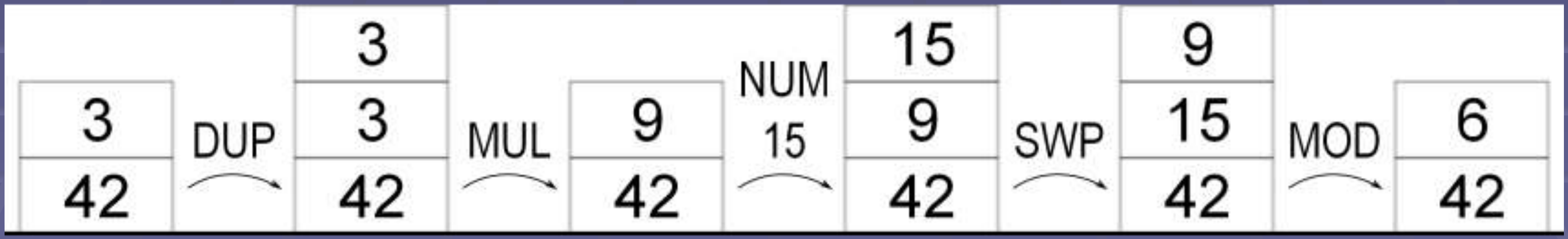

A B C D E F G I L M N O P Q R S T V X Z 1 2 3 4
g k l n p r t v z & x y s o m q i h f d b a c e

Vigenere

Grille

- Pretty easy, wasn’t it?

EXECUTE



3
42
DUP
3
3
42
MUL
9
42
NUM
15
15
9
42
SWP
9
15
42
MOD
6
42

Stack Machine Executor

- Straightforward simulation
- Beware of
  - Integer overflow (MUL)

PROGRAM

Stack Machine Programmer

- The machine language is limited
- Several ways to solve the problem
  - Polynomial
  - Linear combination of some values
  - Implement EQ
  - Implement IF/THEN ☺

Polynomial way

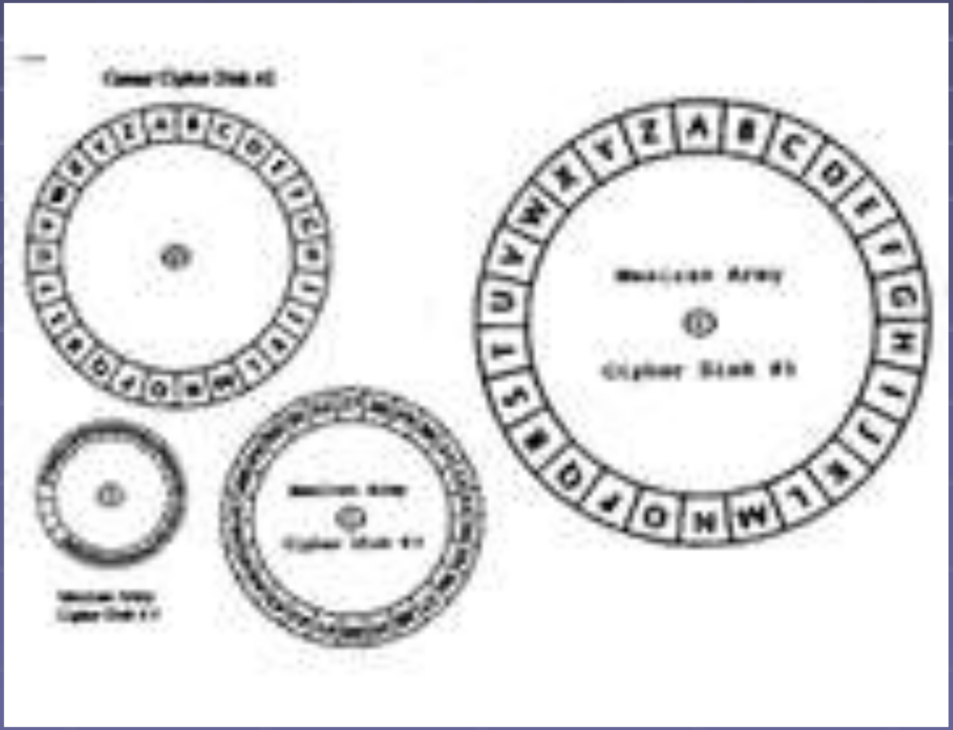
- 1→ 3, 2 → 10, 3 → 20
- Polynomial: $A.x^2 + B.x + C$

- $A.1^2 + B.1 + C = 3$
- $A.2^2 + B.2 + C = 10$
- $A.3^2 + B.3 + C = 20$

“Equals” implementation

- Sort inputs: **2 3 5 8 11**
- Q = (((X mod 11) mod 8) div 5)
  - **Q=1**   iff X=5
  - **Q=0**   otherwise
- Q mul R   (R – desired output for 5)
- Sum for all inputs:
  - $Q_1.R_1 + Q_2.R_2 + Q_3.R_3 + Q_4.R_4 + Q_5.R_5$

ANALYSE



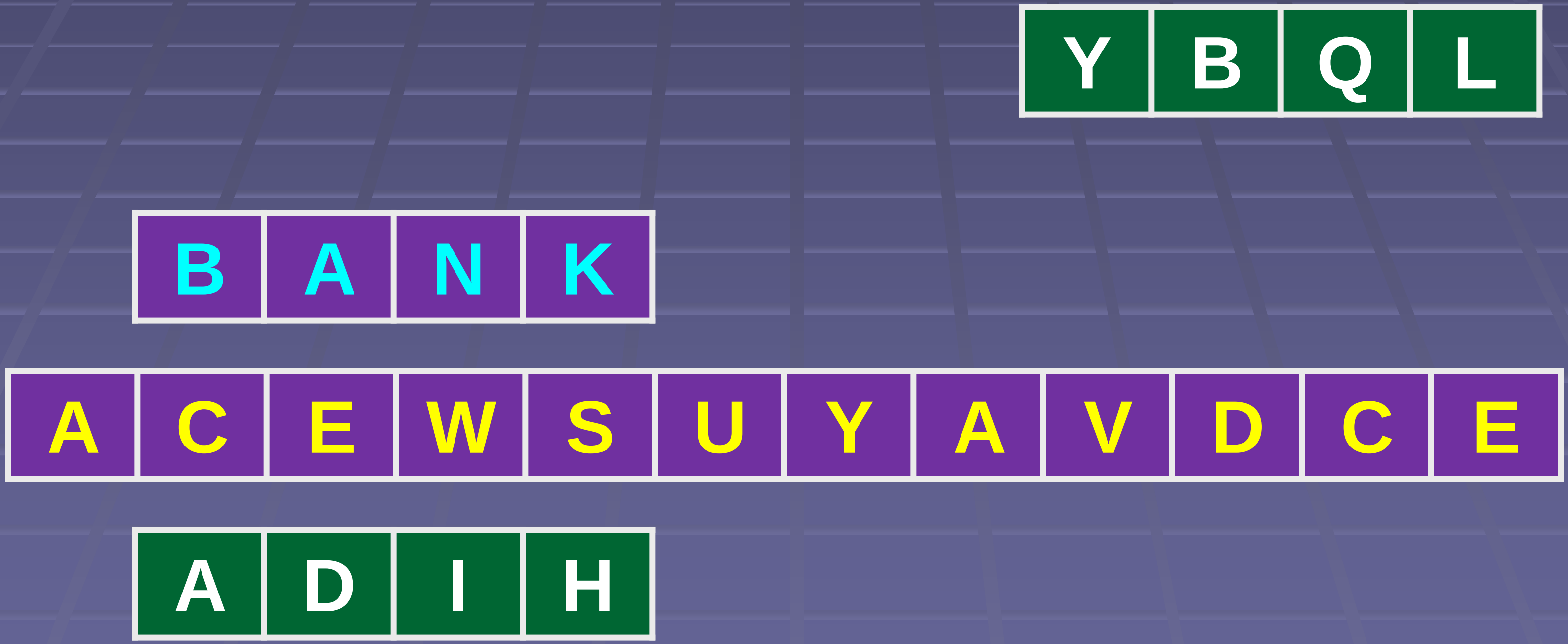
Vigenere Analyse

- We try the cribs in all positions

B	A	N	K								
A	C	E	W	S	U	Y	A	V	D	C	E
Y	B	Q	L								

Vigenere Analyse

- We try the cribs in all positions



Y B Q L

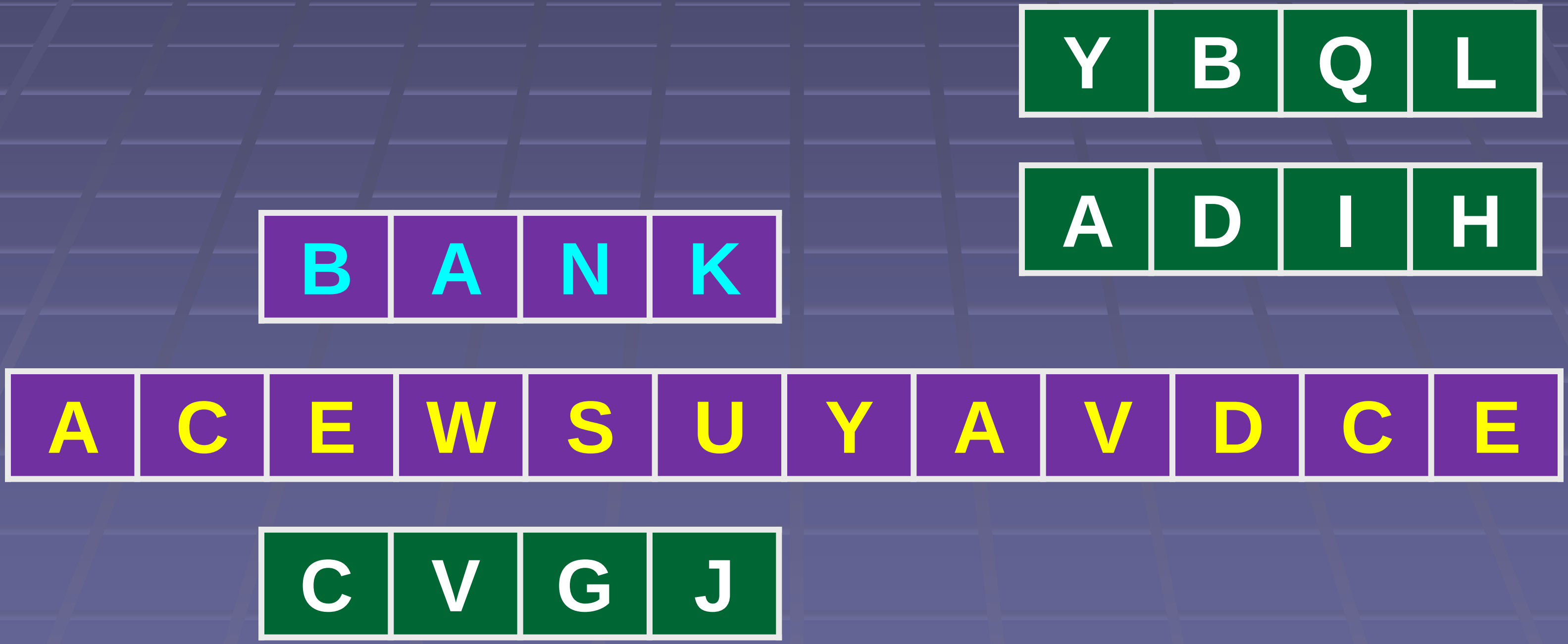
B A N K

A C E W S U Y A V D C E

A D I H

Vigenere Analyse

- We try the cribs in all positions



Y B Q L

A D I H

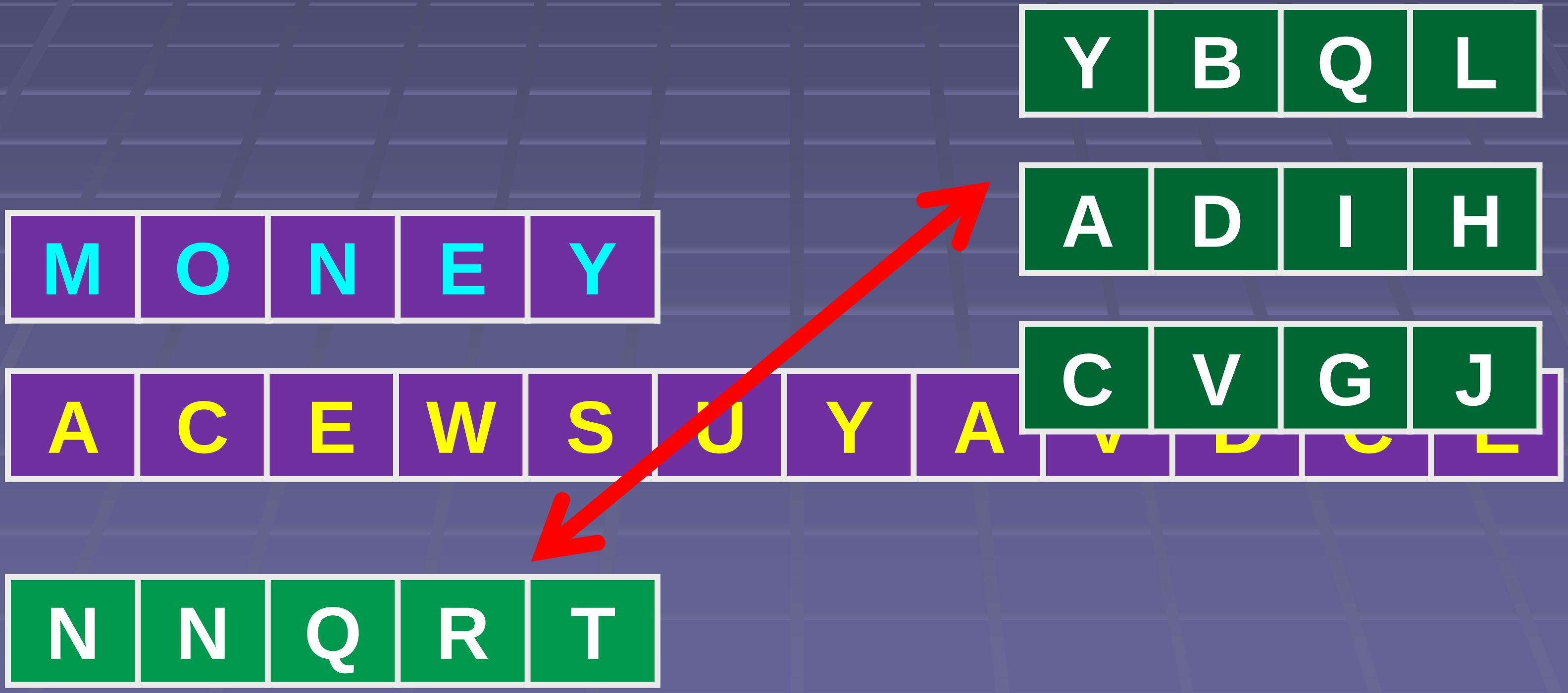
B A N K

A C E W S U Y A V D C E

C V G J

Vigenere Analyse

- We try the cribs in all positions



Y B Q L

A D I H

C V G J

M O N E Y

A C E W S U Y A V D C E

N N Q R T

Analyse

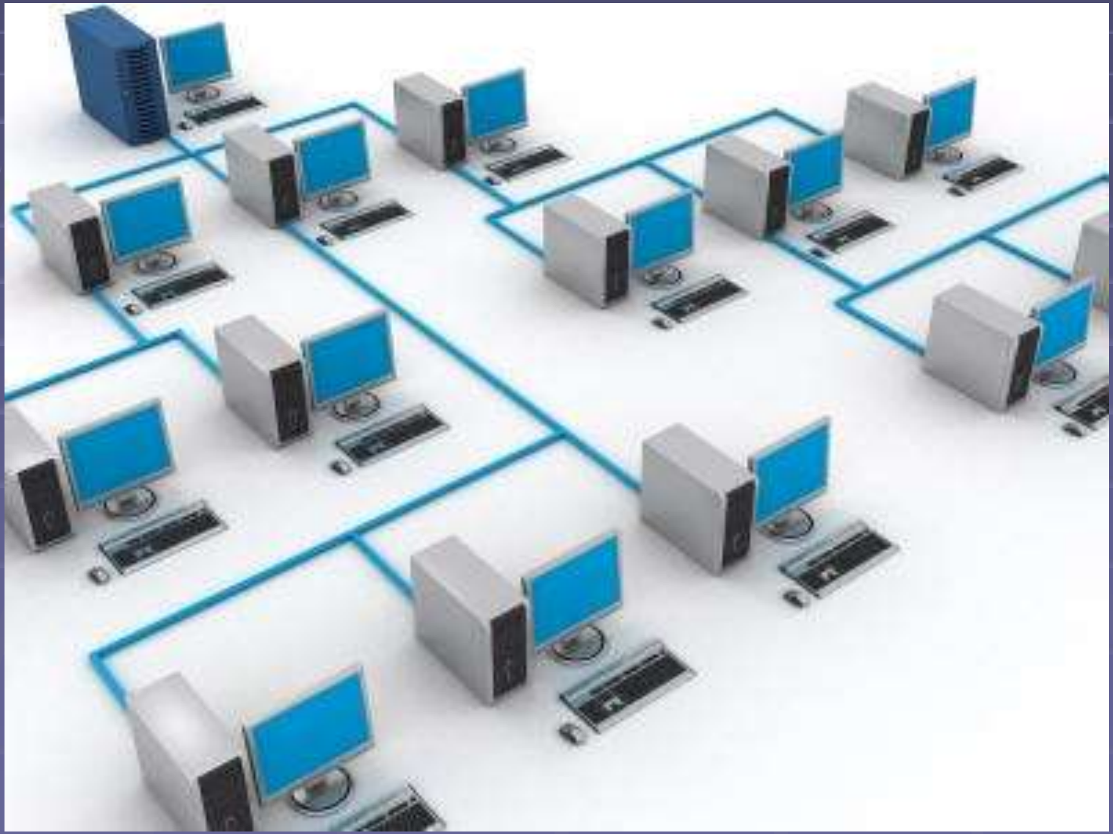
- All placements of the first crib
  - O(n.k)

- All placements of the second crib
  - Test by hash map
  - **O(n.k . H)**

Analyse

- Beware of
  - Key length and repetitions
    - ABCAB → possible keys are ABC, ABCA
  - Overlapping words
    - There should be “two words” in the text
    - Sample input/output had an example

REGULATE

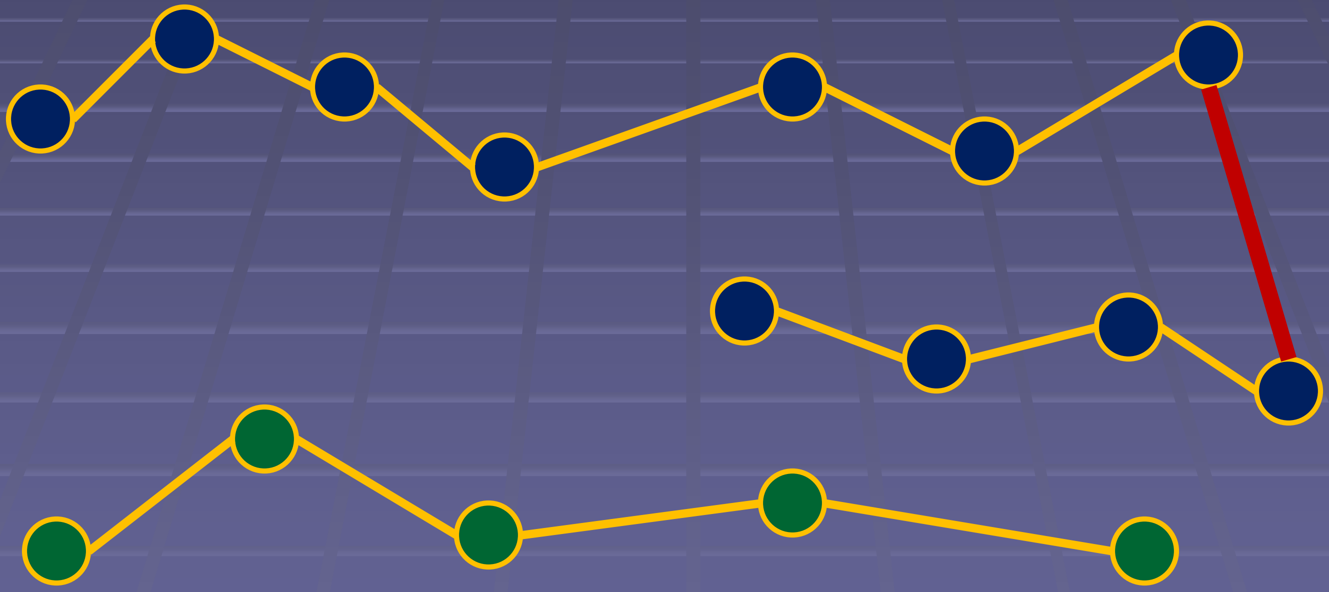


Strange Regulations

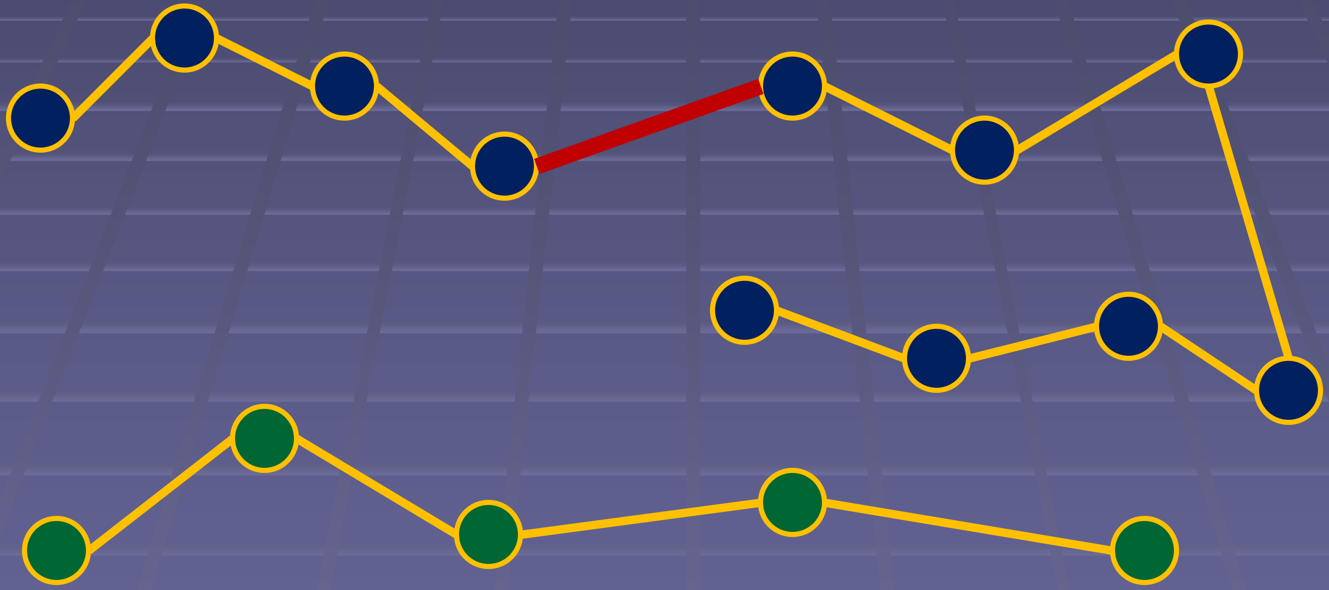
- For each company,
  the cables form linear paths only

- We keep the disjoint-set information
  - find
  - union
  - split

Regulate – Disjoint Sets



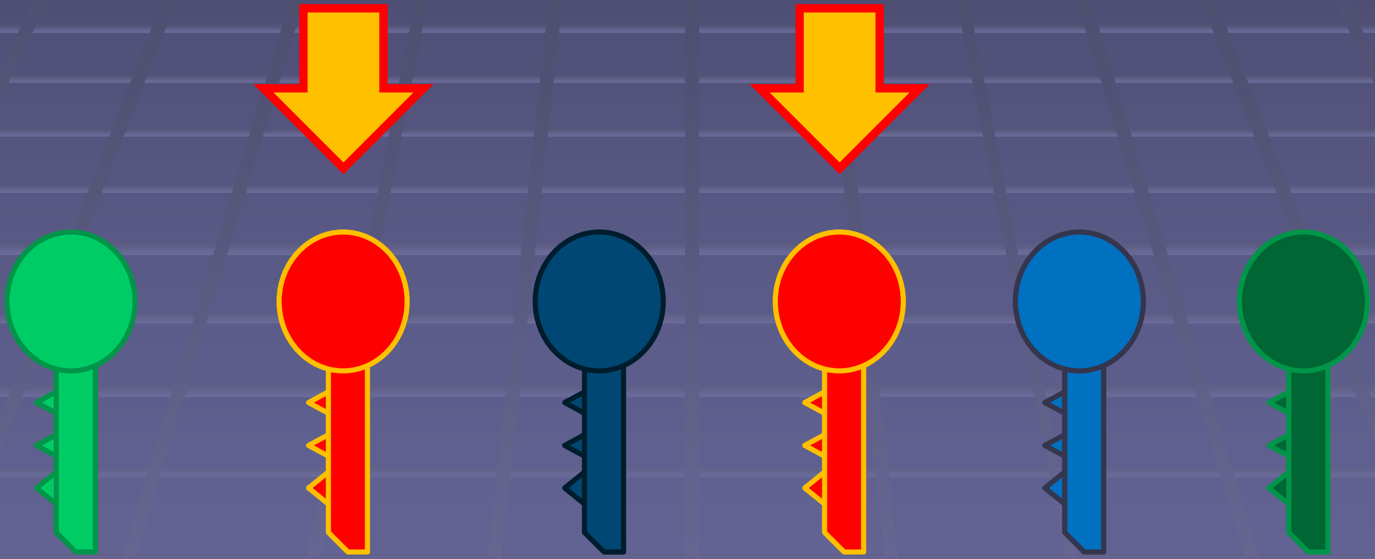
Regulate – Disjoint Sets



Strange Regulations

- We need all operations quickly
  - Tree-based structures
  - **Balancing!!**
- One query
  - **O(log n)**
  - **O(sqrt(n))** – amortized (rebuild)

UNIQUE



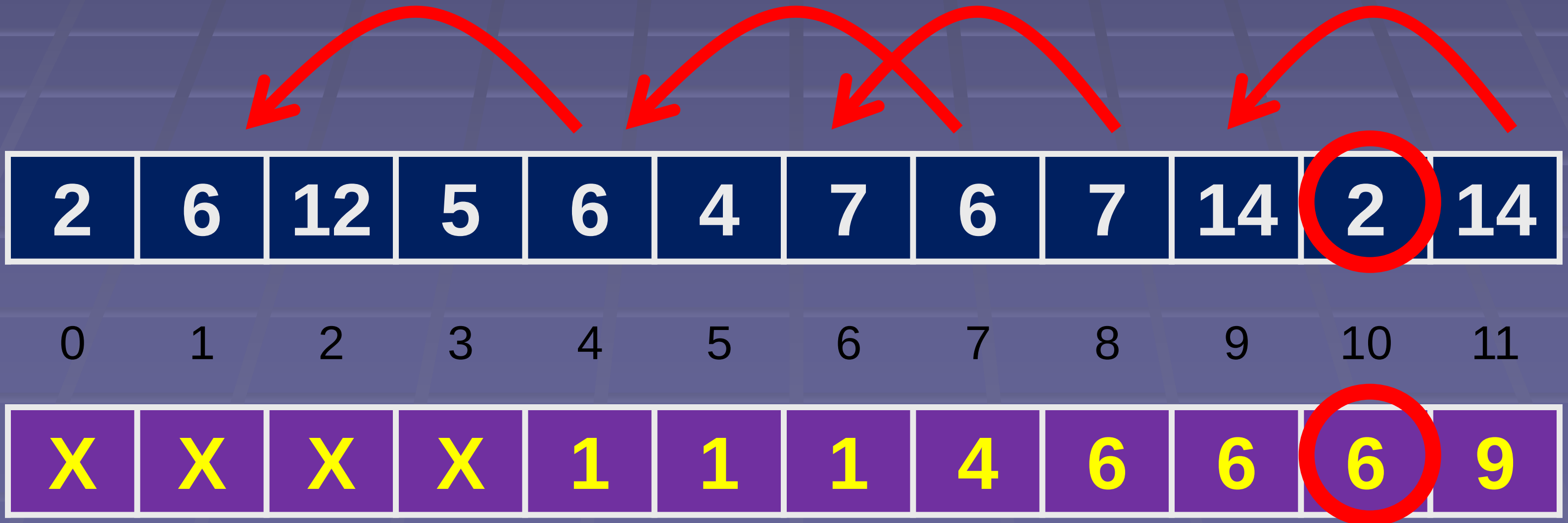
Unique Encryption Keys

- Trivial solution: O(n) for each query

- Prepare a data structure
- Perform the lookups faster

Unique – possible solution

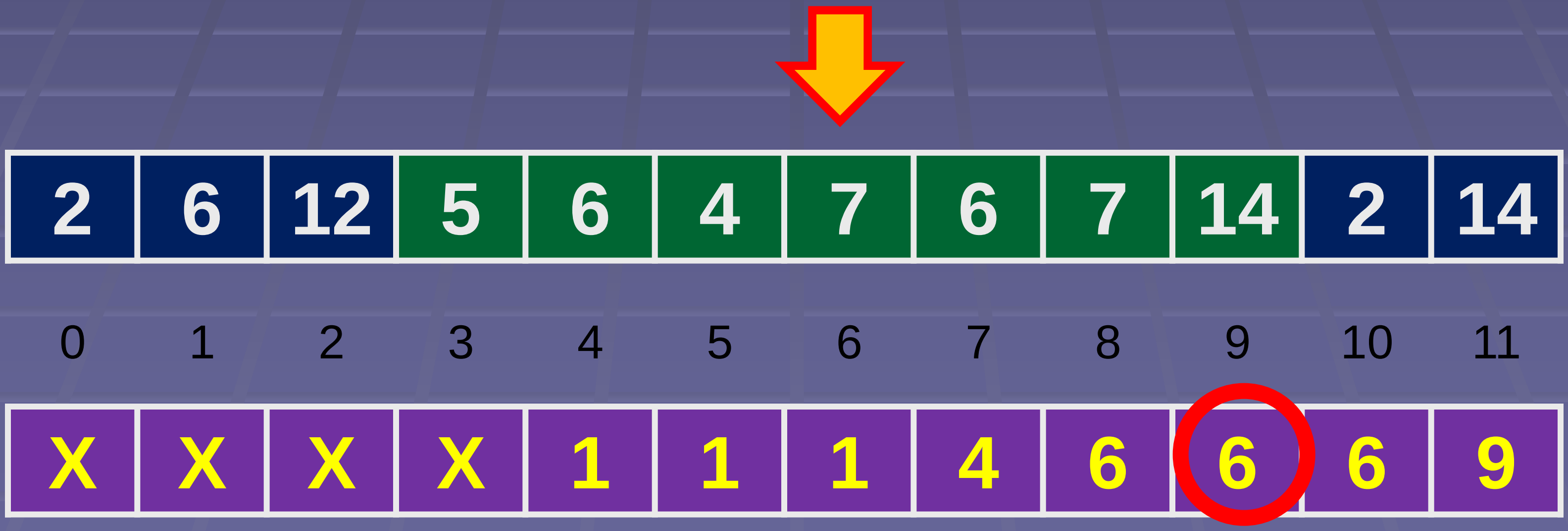
- One possibility:
  - Remember the “last previous” duplicity



2	6	12	5	6	4	7	6	7	14	2	14
0	1	2	3	4	5	6	7	8	9	10	11
X	X	X	X	1	1	1	4	6	6	6	9

Unique Keys

- Query is resolved in O(1)

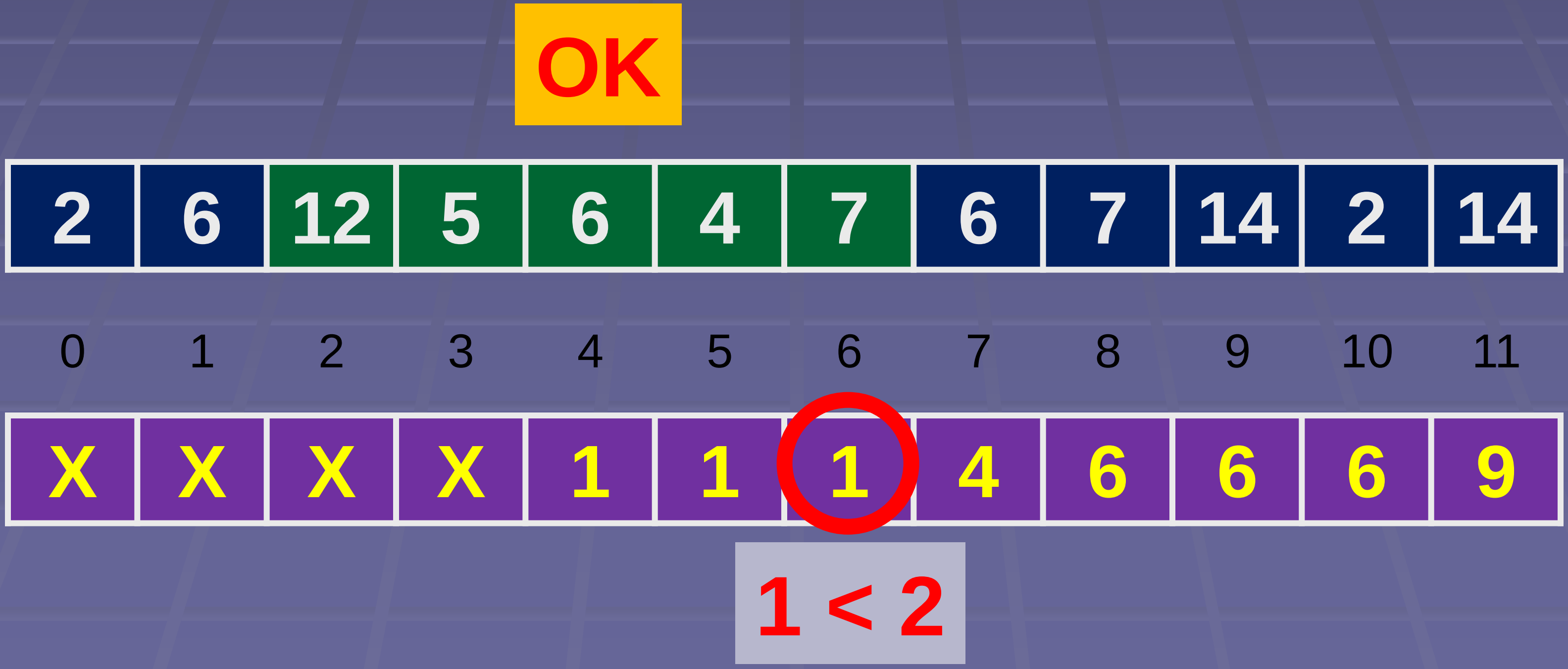


2	6	12	5	6	4	7	6	7	14	2	14
0	1	2	3	4	5	6	7	8	9	10	11
X	X	X	X	1	1	1	4	6	6	6	9

6 ≥ 3

Unique Keys

- Query is resolved in O(1)



OK

2	6	12	5	6	4	7	6	7	14	2	14
0	1	2	3	4	5	6	7	8	9	10	11
X	X	X	X	1	1	1	4	6	6	6	9

1 < 2

Unique – time complexity

- Lookup array prepared: **O(n . log n)**
  - Using a map
- One query: **O(1)**

CARDS

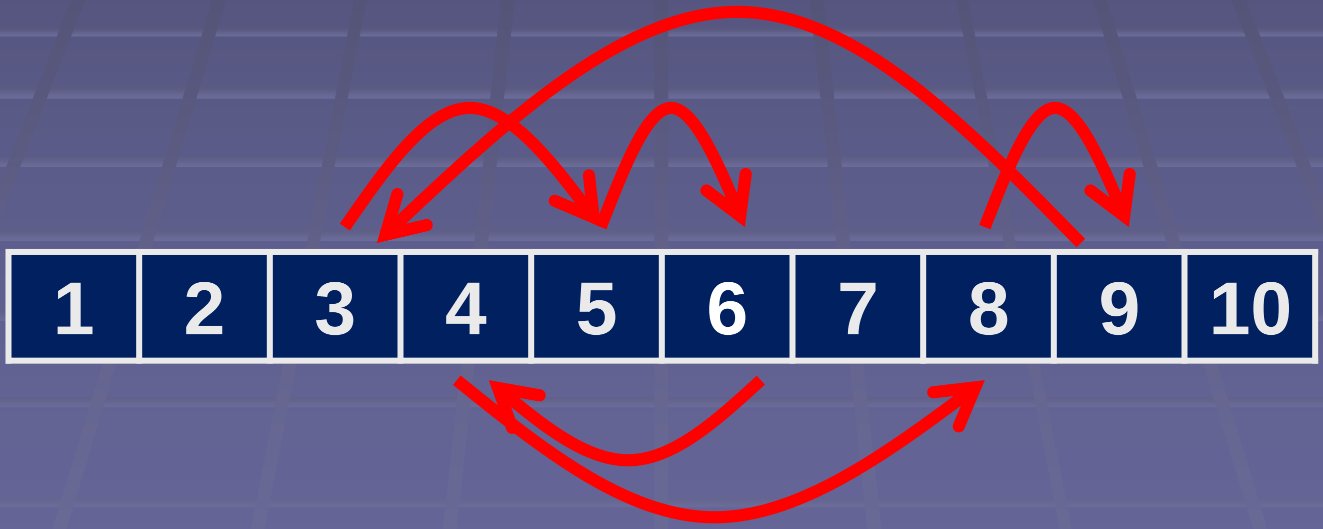
Card Game

- One game = permutation
- Follow the position of all cards

- Each card “travels” in some cycle
  - Periodically repeating occurrences

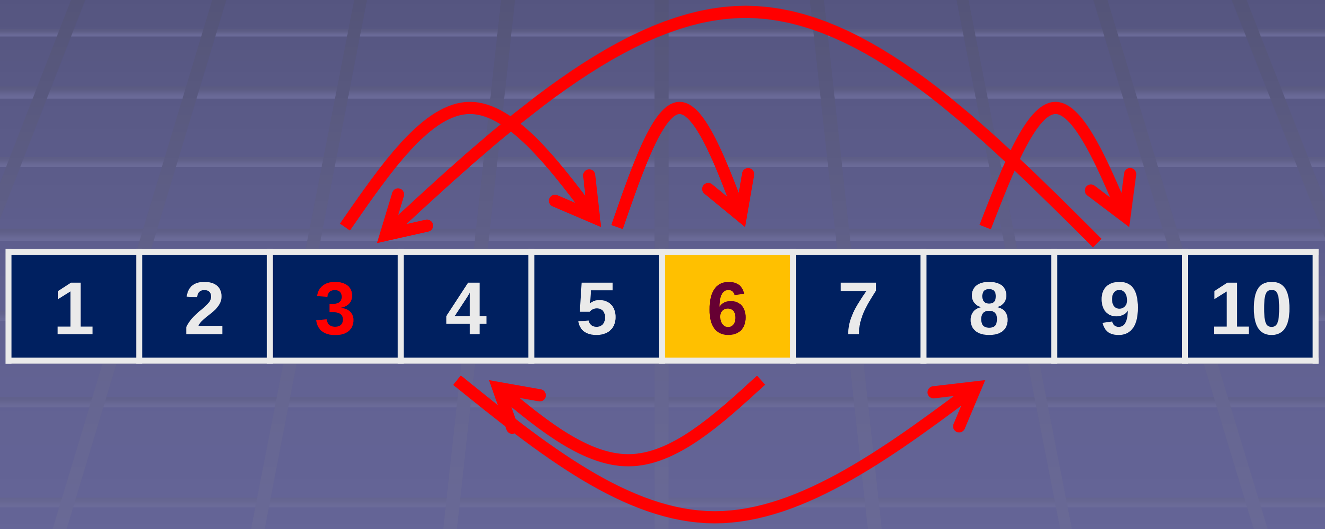
Card Game

- Each card "travels" in some cycle
  - Periodically repeating occurrences



Card Game

- When is the card “3” at position 6?
  - In the game #3 and then every 7th game
  - 7.i + 3



Card Game

- Track all of the cards at all positions
- Card C is at the position P in the deck
  - $F_{CP} + i_{CP}.R_{CP}$
  - never

Card Game

- All winning combinations (120 x N)
  - 1,2,3,4,5,x,x,x,x,x
  - 1,2,3,5,4,x,x,x,x,x
  - 1,2,4,3,5,x,x,x,x,x
  - 1,2,4,5,3,x,x,x,x,x
  - 1,2,5,3,4,x,x,x,x,x
  - … etc.

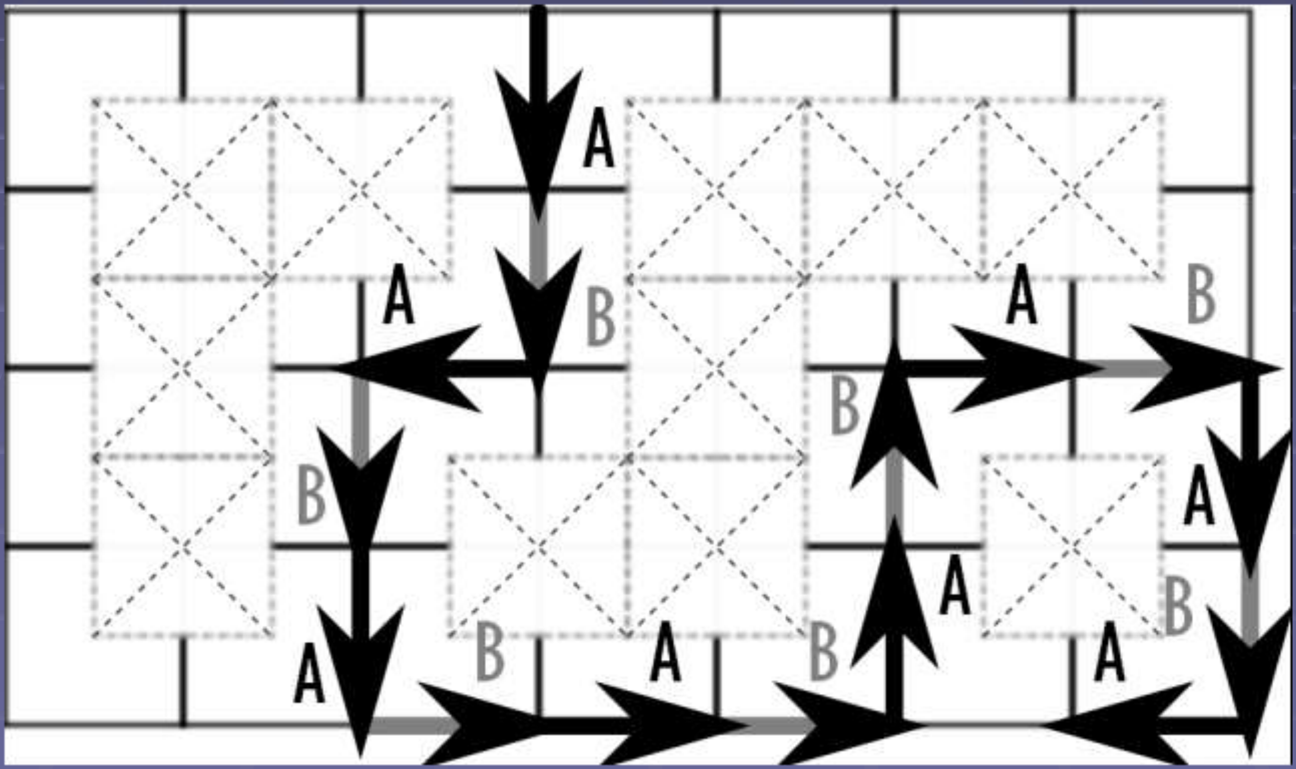
Card Game

- For each winning combination
  - Do the cards ever occur at those places? When?
    - $F_{1P} + i_{1P}.R_{1P}$
    - $F_{2Q} + i_{2Q}.R_{2Q}$
    - $F_{3S} + i_{3S}.R_{3S}$
    - $F_{4T} + i_{4T}.R_{4T}$
    - $F_{5U} + i_{5U}.R_{5U}$

Card Game

- Find the common occurrence
  - Solving the Bezout's identity
    A.i + B.j = C
  - Extended Euclidean algorithm
  - gcd(A,B) divisible by C

TRAIL

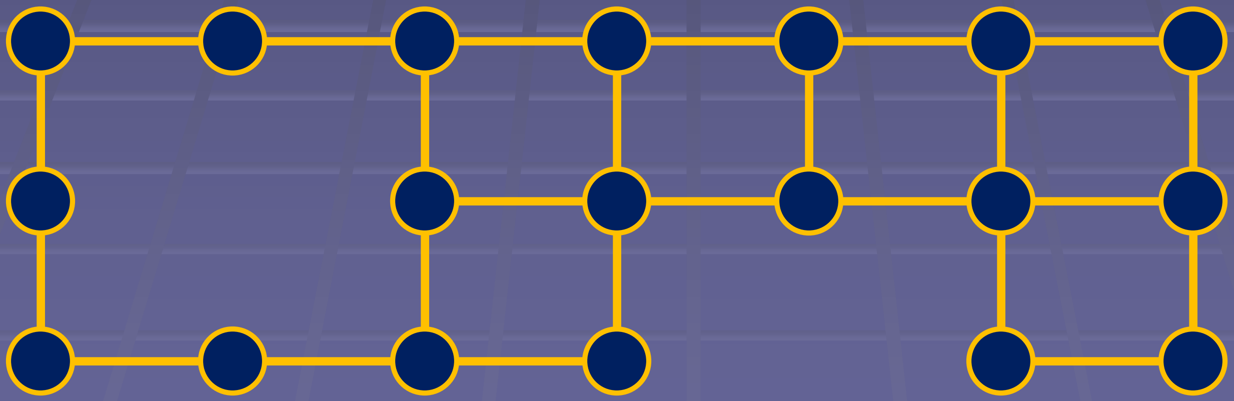


Racing Car Trail

- What we cannot use:
  - Backtracking
  - Dynamic programming
- What to use?
  - Graph theory

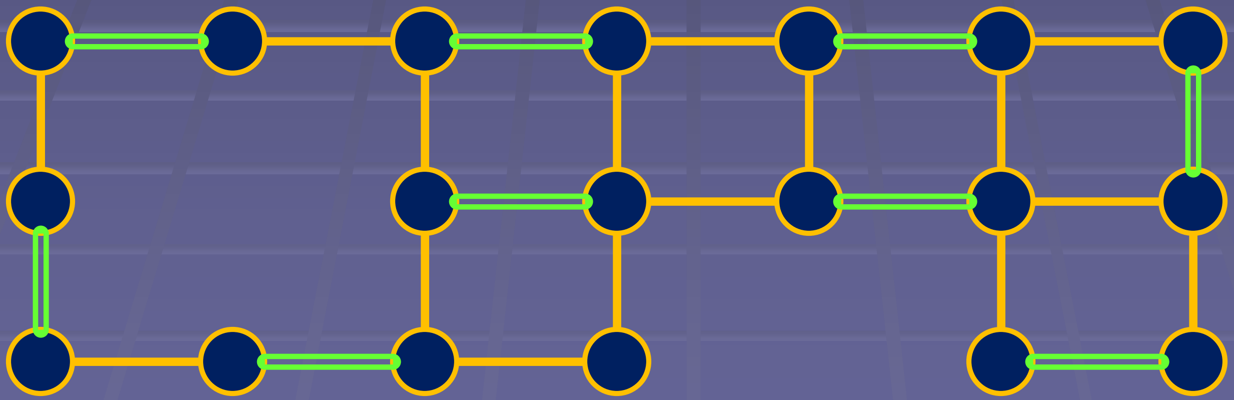
Trail – the graph

- Each position is a node
- Edge if the move is possible



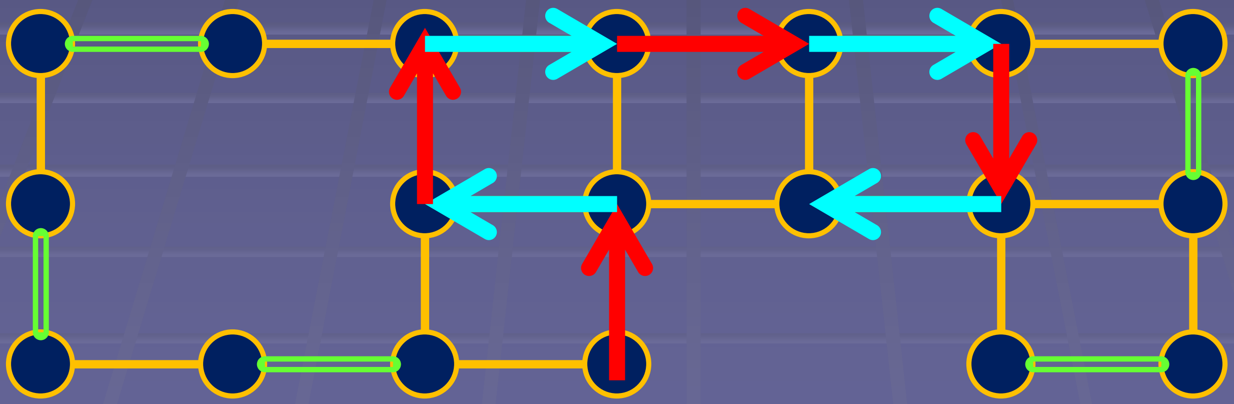
Trail – key observation

- We find the maximum matching



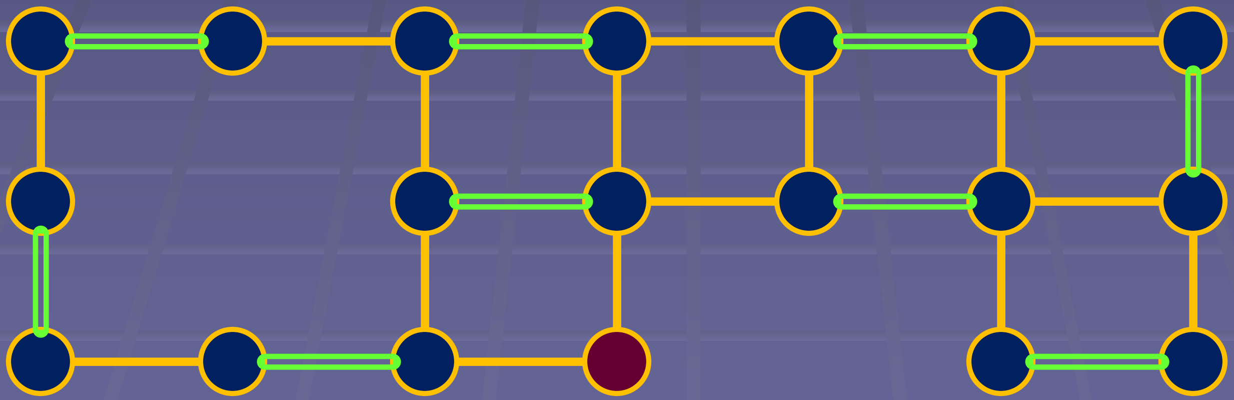
Trail – key observation

- Maximum matching
  - Start from an unmatched node => lose



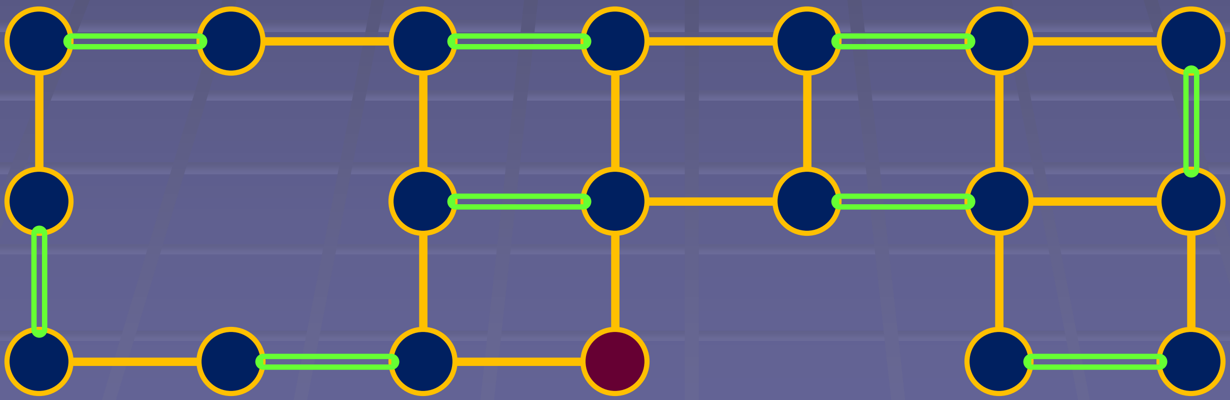
Trail – key observation

- How to find answer to some node?
  - Find maximum matching without it
  - Try to find an augmenting path from it



Trail – key observation

- Does the augmenting path exist?
  - YES => Alice can win
  - NO => Alice will lose

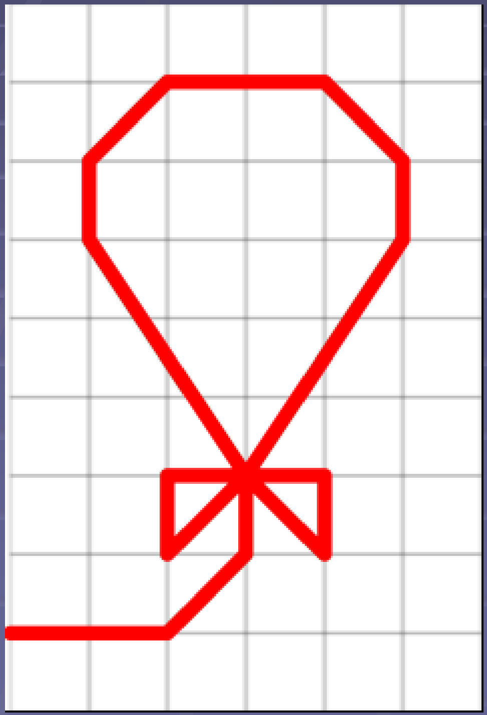


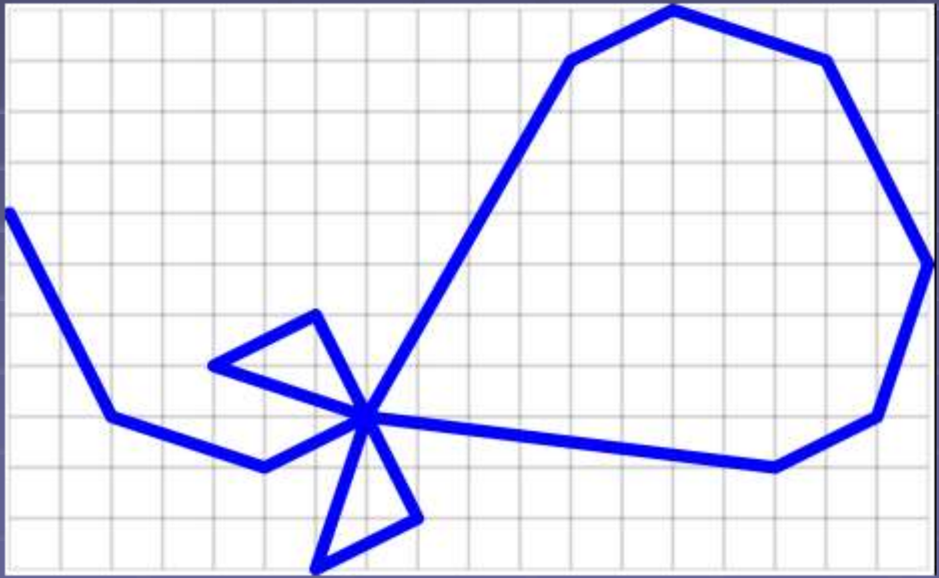
Trail – time complexity

- Turn a matching (without one node) into another by 1 augmenting path

- $O(n^2)$ – the initial matching
- $O(n)$ for each node
- **TOTAL:** $\mathbf{O(n^2)}$

UNCHANGE





Unchanged Picture

1. Picture “normalization”
   - Join overlapping and continuing lines

2. Compare two pictures
   - Try to map one line in Picture 1 to all lines in Picture 2
   - Check if it maps everything

Unchange – time complexity

- Comparing lines – hashing
  - O(n^2 . H)

- O(n^3) is too much!

Unchange – faster solution

- Find the “center of mass” X
- Points in the longest distance from X map to each other
    - “Tie-breakers”

- Not required in this contest (1000 lines max)

Authors

Josef Cibulka

Jakub Černý

Zdeněk Dvořák

Martin Kačer

Jan Stoklasa

Jan Katrenic

Radek Pelánek