# Day 1: Problem Analysis

Maxim Buzdalov

ITMO University

10.03.2015

# Preliminaries

- Contest origin – German Collegiate Programming Contest 2010
- Original winner – U Erlangen: deFAUlt, 10/11 problems solved
  - $7^{th}$ place at World Finals 2011
  - thought this contest is more-or-less difficult
- Pre-solving: Scala, bytes of code / problem

| | | | | | |
|---|---|---|---|---|---|
| A | 715 | E | 2009 | I | 802 |
| B | 2154 | F | 3001 | J | 1895 |
| C | 789 | G | 1166 | K | 1711 |
| D | 1968 | H | 1595 | | |

- Result: an easy contest

# Problem A. Absurd Prices

- Given a price
  - an integer $1 \le c \le 10^9$
- Determine if it is absurd
  - $c = XX\ldots Xd000\ldots00$
  - $a =$ number of $X$s
  - absurdity: $2 \cdot a - ($ if $d = 5$ then 0 else 1
  - $c$ is absurd <=> $[0.95 \cdot c; 1.05 \cdot c]$ contains an integer with a smaller absurdity

# Problem A. Solution

- What is the nearest integer with a smaller absurdity?
- Case 1: $c = XX\ldots X500\ldots 0$
  - Try replacing $X5$ with $X0$ or $(X+1)0$
- Case 2: $c = XX\ldots X(d \neq 5)00\ldots 0$
  - Try replacing $Xd$ with $X0$, $X5$ or $(X+1)0$

# Problem B. Cheating or Not

- Distribute soccer teams into groups

| A | B | C | D |
|---|---|---|---|
| ITA | URU | IRE | ARG |
| YUG | GER | CZE | ENG |
| NET | BRZ | SPA | ROM |

- Constraints
  - first row is fixed
  - some teams cannot be in the same group
  - otherwise rows are random
- Each team has a strength
- Find expected sum of strengths of team $T$'s competitors

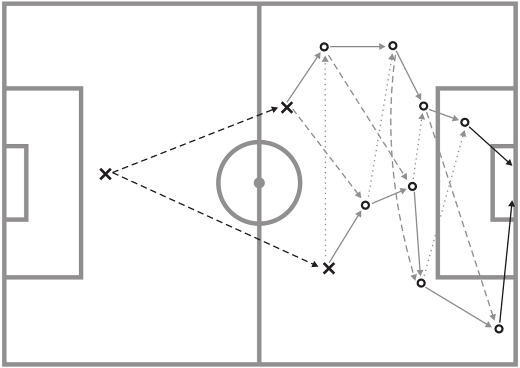
# Problem B. Solution (1/2)

- A key insight:
  - no two teams from the same "confederation" can be in the same group
  - apart from the first row, all confederations fit into rows
    - confederations bigger than a row are ignored
  - no two rows $2 \le i < j \le m$ influence each other!
- Compute separately:
  - probabilities for the desired team to appear in each group
  - expectations of strength sums (not including the desired team's row) for each group
  - the answer is the scalar product of the two

# Problem B. Solution (2/2)

- For each row:
  - If it contains the desired team:
    - try all row permutations
    - filter prohibited ones
    - compute probabilities for each group to appear in
  - Otherwise:
    - try all row permutations
    - filter prohibited ones
    - compute expected strengths
- Consider the seeded row separately:
  - If it contains the desired team:
    - set the corresponding probability to 1.0
  - Otherwise:
    - add strengths to expectations

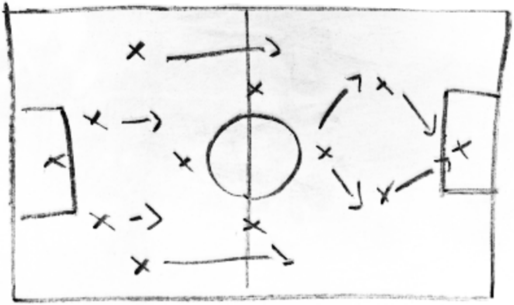
# Problem C. Counterattack



# Problem C. Counterattack

- Given:
  - source $S$ and target $T$
  - $2N$ more vertices: $P_1 \dots P_N$, $Q_1 \dots Q_N$
  - costs: $\{P, Q\}_i \to \{P, Q\}_{i+1}$, $S \to \{P, Q\}_1$, $\{P, Q\}_N \to T$
- Find the shortest path from $S$ to $T$
- Solution: the graph is acyclic $\to$ DP, $O(N)$ time

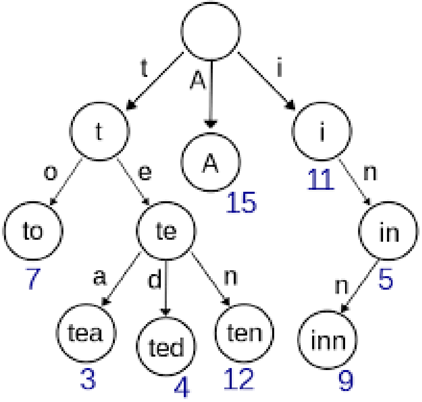
# Problem D. Field Plan



# Problem D. Field Plan

- Given $N$ positions on the field and $M$ possible moves
- Find all positions from which all other positions are reachable using possible moves
- Solution:
  - Find the topologically first connected component – $O(N+M)$
  - Check if all vertices are reachable from it – $O(N+M)$
  - If not, `Confused`, otherwise print the component

# Problem E. Hacking



# Problem E. Hacking

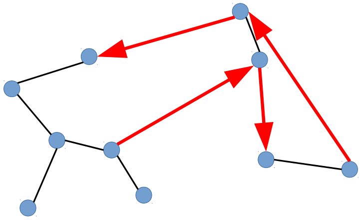
- Given:
  - a string $s$ of length $n \leq 10\,000$
  - a maximum query length $m \leq 100$
  - a limit on the alphabet to use $k \leq 26$
- Find a query which is not a substring of $s$
- Solution
  - build a trie from all substrings of length $\leq m$
  - traverse using symbols $1 \leq s \leq k$
  - find a non-existing node at depth of at most $m$ and report a path to it

# Problem F. Last Minute Constructions

- Given:
  - $N$ vertices, $M$ undirected edges, a forest
  - $T$ more directed edges
  - a starting vertex $S$ and a finishing vertex $T$
- Find:
  - a path from $S$ to $T$
  - ... traversing each directed edge in the correct direction
  - ... which may use undirected edges
  - ... visiting each vertex at most once
  - ... or determine it is impossible

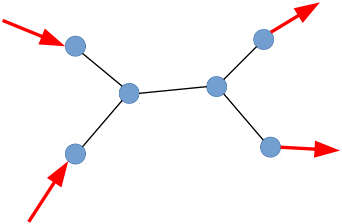
# Problem F. Solution Idea

- Add one more directed edge: from $T$ to $S$
- Looks like one needs to find an Eulerian cycle
  - edges $\leftarrow$ directed edges
  - vertices $\leftarrow$ components connected by undirected edges



# Problem F. Complication

- However, not every incoming edge can be connected to every outgoing edge
- Remember, each vertex can be visited only once!



# Problem F. Solution

- In each connected component:
  - it is a tree
  - determine which incoming edges can be connected to which outgoing edges – dynamic programming
  - if mapping exists, it is the only one
- After connecting incoming edges to outgoing ones:
  - find Eulerian cycle
  - simpler: it's a loop, so just check connectivity

# Problem G. Lineup

- Given:
  - 11 players, 11 positions
  - $A(player)(position)$ – profit
  - For each player, profit is non-zero for at most 5 positions
  - Find an optimum assignment
- Solutions:
  - bipartite matching
  - Hungarian algorithm
  - brute-force in $11^5$
  - ...

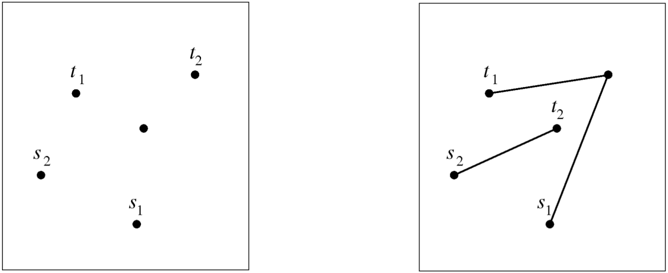
# Problem H. Polynomial Estimates

- Given: $x_1, \ldots, x_N$
- Determine: if $x_i = P(i)$ for some polynomial $P$ with degree of at most 3
- Solution:
  - Complex: compute the polynomial from first 4 values, check all other values
    - big integers
    - or fighting with precision
  - Simple:
    - compute neighbor differences of the input
    - compute neighbor differences of the above
    - compute neighbor differences of the above
    - if they are all same, it's a polynomial with degree of at most 3

# Problem I. Soccer Bets

- Given: shuffled olympic game outcomes without draws (+ a game for $3^{\text{rd}}$ place)
- Find: the winner
- Solution: find who lost no games

# Problem J. The Two-Ball Game



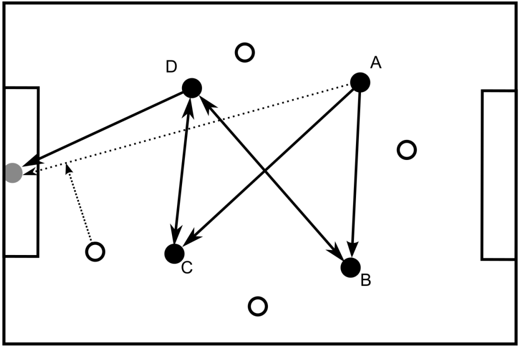
# Problem J. Idea

- Assume that the game is possible, but it is not possible if $S_1 \to T_1$ directly
  - consider convex hulls $H_A$ of points above $S_1$–$T_1$ and $H_B$ of points below
  - $H_A \cup H_B \cup \{S_1, T_1\} =$ all points
  - no point of $H_A$ can see no point of $H_B$
  - then the path from $S_1$ to $T_1$ should be either above $S_2$ or below $T_2$
  - the path from $S_1$ to $T_1$ can be convexified, then both $S_2$ and $T_2$ will be strictly inside
  - then it's possible to use $S_2 \to T_2$ directly!

# Problem J. Solution

- If $S_1$–$T_1$ doesn't intersect $S_2$–$T_2$, then `POSSIBLE`
- Check if it's possible using $S_1$–$T_1$, then using $S_2$–$T_2$
- How to perform each check?
  - maximum, minimum are by $(p - S_1) \cdot (T_1 - S_1)$
  - $P_A^+$ – maximum from points above $S_1$–$T_1$
  - $P_A^-$ – minimum from points above $S_1$–$T_1$
  - $P_B^+$ – maximum from points below $S_1$–$T_1$
  - $P_B^-$ – minimum from points below $S_1$–$T_1$
  - if $P_A^+$–$P_B^+$ doesn't intersect $S_1$–$T_1$, it's OK
  - if $P_A^-$–$P_B^-$ doesn't intersect $S_1$–$T_1$, it's OK

# Problem K. To Score or Not to Score



# Problem K. To Score or Not to Score

- Given:
  - robot positions (black team, white team) and the goal
  - robots may intercept a ball
  - a robot is 3 times slower than a ball
  - each robot except for the starting one may be knocked out
- Find if it is possible to score a goal?
- Solution:
  - build a “passability” graph
  - try knocking out all vulnerable vertices
  - check if the goal is reachable nevertheless