2026 NWERC

NWERC 2026 Test Session

Solutions presentation

November 28, 2026



KIT Karlsruhe Institute of Technology

CPUlm

icpc International Collegiate Programming Contest

A: Assembling Primes

Paul Wild

Problem

You are given a set of digits. Construct a prime number that uses exactly those digits, possibly with repeats.

Solution 1 – Randomization

- Generate 1000 random numbers using the given digits
- Primes are quite frequent, so if there exists a solution you will find it
- An $\mathcal{O}(\sqrt{n})$ primality test should be fast enough

A: Assembling Primes

Paul Wild

Solution 2 – Systematic search

- First, some case analysis:
  - The only primes using just one digit are 2, 3, 5, 7 and 11
  - If the GCD of the digits is ≥ 2, no solution exists
  - If none of 1, 3, 7 or 9 are in the set, no solution exists
  - A solution exists in all other cases
- Recursively create all valid strings in order (or use Python's `itertools`) until you find a prime.
- Pitfalls: don't put 0s at the front; be careful with large sets whose digit sum is a multiple of 3

Solution 3 – Precomputation

- If your solution is a bit too slow, just compute answers for all 1023 possible cases locally
- Submit a simple program that outputs the precomputed answers

Statistics: ... submissions, ... accepted, ... unknown

B: Bicryption

Yidi Zang

Problem

Implement a winning strategy for the following encryption/decryption protocol:

- Anna and Ben are each given a word
- To encrypt a word, they need to replace each letter by some different letter
- They each encrypt their word twice, using two distinct letters in each position
- Charlotte receives the encrypted words from one of the players, in arbitrary order
- She then needs to identify the person, as well as their original word

B: Bicryption

Yidi Zang

Solution

- Anna Caesar shifts her word by ± 1 and Ben by ± 2
- To identify the sender, Charlotte checks if the distance of first letters is ± 2 or ± 4
- To decrypt, she Caesar shifts by half that distance

Many other solutions are possible.

Statistics: ... submissions, ... accepted, ... unknown

C: [kœri]wurstwerfen

Maarten Sijm

Problem

Compute the maximum possible time a [kœri]wurst can be airborne.

Solution

- The problem statement gives a formula to compute the height difference in terms of the time
- Rearrange and solve this equation for time in terms of height difference
- This only accounts for half the airtime, so multiply by 2
- Pitfall: don't read the input into a 32-bit integer
- Non-pitfall: somewhat unusually, using 32-bit floats is actually fine in this problem (why?)

Statistics: ... submissions, ... accepted, ... unknown