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.

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

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

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