Computer Science Team Week 8

Henry Gustafson Julian Bauer

The College Preparatory School

Computer Science Team
January 22, 2024

What is USACO?

- Coding competition
- Next contest this weekend (Friday-Monday)
- Can go International Olympiad in Informatics if good

Divisions

- Bronze (everyone starts here)
- Silver
- Gold
- Platinum

Scoring

- 4 hours to complete 3 problems
- Each problem is 333 points
- Points based on how many test cases you pass (10-20) test cases, later test cases harder
- Score = (333 * % of 1 tests) + (333 * % of 2 tests) + (333 * % of 3 tests)
- If perfect score you get immediately promoted to next division, else if above 700-800 you get promoted in next contest

Resources

- Long video about USACO
- Official USACO website
- Official instructions/rules
- USACO practice website

Fun coding problems

Theme: Linguistics Olympiad!

Problem Flick

Problem Flick Martians have spooky number system and spookily want to xor it together spookily. Given two strings of 1s and 0s, create a function that applies the xor operation on those two strings and returns a new string of 1s and 0s.

```
def xor(
    right: str, left: str
) -> str
```

Problem Flick

Example

```
assert xor([
    '10110',
    '01010',
]) = '11100'
```

Problem Bengalese Finch

Problem Bengalese Finch One important feature of human language is unpredictability, differentiating it from more predictable sound sequences like birdsong. One simple way to measure this predictability is to measure the chance of one letter coming after another—for example, in English, u comes after q almost 100% of the time. Given a list of words, for each of the letters, create a dictionary of the likelihoods of other letters coming after them from 0.0 to 1.0. Return another dictionary that associates the likelihood dictionaries with their letters. NACLO problem for more fun!

```
def likelihoods(
    words: list[str]
) -> dict[str, dict[str, float]]
```

Problem Bengalese Finch

Example

```
assert likelihoods(
    ['hola', 'oh']
) == {
    'h': {
        'o': 1.0,
    },
    'o': {
        'l': 0.5,
        'h': 0.5
    },
    'l': {
        'a': 1.0
    },
    'a': {}
}
```

Problem Rotodromes

Problem Rotodromes Given a list of all words in the English language and the integer *n*, find out which words, when longer than *n* and rotated to the right by *n*, are still English words. Thank you Stevie W

```
def rotodromes(
    words: list[str],
    n: int
) -> list[str]
```

Problem Rotodromes

Example

```
assert rotodromes([
    'demo', 'mode', 'windup', 'upwind', 'gimkit'
]) == ['demo', 'mode', 'windup']
```

The End

Questions? Comments? Remarks?
Considerations? Confusions?