Computer Science Team Week 4

Henry Gustafson Julian Bauer

The College Preparatory School

Computer Science Team
November 6, 2023

ACSL when?

Friday, November 17th good day?

Review of topics

1. Number bases
2. Recursion
3. What does this program do?

Link to study materials at monad.rocks/acsl

Fun coding problems

Theme: music!

Problem Sadness

Problem Sadness Jofferey is listening to sick beats containing only major triads. However, Jofferey recently died to a laser in *Gimkit: Don't Look Down* and feels sad, so they want to listen to music containing only minor triads. Given a list, *chords*, containing tuples of three numbers that each represent a major chord, output a list of minor chords. You can convert a major chord to a minor chord by subtracting one from the middle number.

```
def sadnessify_music(
    chords: List[Tuple[int, int, int]]
) -> List[Tuple[int, int, int]]
```

Problem Sadness

Example

```
assert sadnessify_music([
    (2, 6, 9),
    (4, 8, 11),
]) = [
    (2, 5, 9),
    (4, 7, 11),
]
```

Problem Sleep

Problem Sleep Joanne the evil robot has arranged a row of 10,000 drums with a different loudnesses, and she is banging ranges of them to prevent you from sleeping. Luckily, you have noise-cancelling headphones, but they require the total amount of loudness to cancel accurately. Given *loudnesses*, a list containing each drums' loudnesses in order, and *bangs*, a list of tuples, each of which is the inclusive range of drums Joanne banged at that instant, output a list of total loudnesses for each bang. **Extra fun:** Make it fast for 100,000 bangs.

```
def total_loudnesses(
    drums: List[int],
    bangs: List[Tuple[int, int]]
) -> List[int]
```

Problem Sleep

Example

```
assert total_loudness(
    [1, 2, 3, 4], # imagine this is 10,000 long
    [(0, 0), (0, 3), (2, 3)]
) == [1, 10, 7]
```

Problem Sevish

Problem Sevish Your favorite musical artist Sevish wants to make his music more beautiful. He thinks chords are *beautiful* when the frequency ratios of their notes can be expressed as simple fractions, which he defines as any fraction where the numerator and denominator are < 16. Given *chords*, a list of three-note chords where each chord is a list of floating point frequencies, output a new list of lists of integers < 16 corresponding to each chord, where for each chord the ratios of between the integers correspond as closely as possible to the ratios between the original frequencies. **Extra fun:** Make it work for 10-note chords.

```
def sevishify(chords: List[List[float]])
    -> List[List[int]]
```

Problem Sevish

Example

```
assert sevishify([
    [1.0, 2.0, 3.015],
    [36.0, 17.9876, 108.11]
]) == [
    [1, 2, 3],
    [2, 1, 6]
]
```

The End

Questions? Comments? Remarks?
Considerations? Confusions?