CS 61A Challenge Problems: Mutable Data

Solutions at http://alextseng.net/teaching/csm61a/

Alex Tseng

---

1 Environment Diagrams of Lists

Draw the environment diagrams of the following. Assume execution is all in the global scope.

(a)

```
s = [[1, 2], [3, 4]]
t = s[1]
```

(b)

```
s = [[1, 2], [3, 4]]
t = s
t[1][1] = list(s)
```

(c)

```
s = [[1, 2]]
t = list([s, s])
t[1] = list(s)
```

2 Linked Lists

(a) Create a linked list that includes a loop. That is, if we were to continuously call `rest` on the list, we would never reach `"empty"`.

(b) *Challenge* Write a function `has_loop(s)` that checks if `s` has a loop. Pseudocode is fine, but make sure you can translate it into native Python.

3 List and Dictionary Comprehensions

(a) Using a single (possibly nested) list comprehension, compute the set of prime numbers from 0 to 99 (inclusive). Your list comprehension should return a list of lists, where the `i`th list is the list of prime numbers in `[i*10, (i*10)+9]`. The result should look something like:
`[[2, 3, 5, 7], [11, 13, 17, 19], ...]`
You may assume that there is a function `is_prime(x)` that returns `True` if `x` is prime and `False` otherwise.

(b) Use a single dictionary comprehension that maps each element of a list `items` to the number of times it appears in `items`, but only if it appears more than 2 times.
If `items` is: `["A", "A", "A", "B", "B", "C", "C", "C", "C", "D"]`,
then the result will be: {`"A": 3, "C": 4`}

(c) Use a single list comprehension to compute the set of right triangles with *integer* side lengths no more than 30 (each side must be an integer ≤ 30). A triangle is defined by its three sides. Your list comprehension should return a list of tuples, each with the lengths of the three sides:
`[(3, 4, 5), (5, 12, 13), ...]`
Hint: all right triangles follow the Pythagorean theorem.