

Lecture 20 (3-2-26)

Exam 02 Review: Python

- if point total is 0.3, then there are 3 right choices
 - more fill in the blanks on this exam
 - remember you should sort by lowest priority first and highest priority last
-

1.5 pts Python Scripting

- will be given a script with bugs, and you'll have to fix it
-

1.6 Data Processing

- Given some JSON data structure what does each statement do
 - there could be `indexError`, `TypeError`, `keyError` (so know these)
-

1.5 pts Functional Programming

- you'll write some specified function in 3 different ways
 - also know:
 - What is the memory tradeoff?
 - How is `map/filter` different from a list?
 - How to use a Python script as a module?
 - How to write a doctest
 - how to annotate types for functions
 - how to run doctest
 - THERES one more, watch panaptop
-

1.5 List Comprehensions

- What is the memory tradeoff
- How is list comprehension different from a list?

- How to implement and run doctest?
-

1.5 Generators

- What is the memory trade-off?
 - How is generator different from a list?
 - How to generate a sequence of numbers?
 - this is range() so review range
-

1.4 pts Concurrency and Parallelism

- Given something
 - identify if data parallelism
 - identify if task parallelism
 - identify if embarrassingly parallel
 - so know these three terms
 - task parallelism : you can switch back and forth between tasks?
 - remember generators run something, yield, and run something else
 - to remove task parallelism, you could switch all generator expressions to list comprehensions
 - data parallelism: run the exact same function different parts of my dataset
 - so `map()` and `filter()` is used for this
 - by default in python you would use only 1 core
 - so concurrent but not parallel with 1 core
 - if you use `concurrent.futures.ProcessPoolExecutor()`
 - then you are doing parallelism
 - Embarrassingly parallel: splitting up the labor
 - if program is data parallel then it is usually always embarrassingly parallel
-

2.0 pts Translations

- do two translations
 - unix pipeline to python code
- not meant to be literal

- so no `os.systems()`
 - no `os.popen()`
 - `cat -> open()`
 - regex in python -> `re.findall()`
 - review how to initialize set
-

```
#making a pipeline

ps aux | awk '{print $1}' | sort | uniq -c | sort -rns

# -s in the last sort is for `stable sort`
```

```
#!/user/bin/env python3

import os

for process in os.popen('ps aux'):
    print(process.split()[0])

# equivalent:

# users = [process.split()[0] for process in os.popen('ps aux')]

counts = {}

for user in users:
    counts[user] = counts.get(user, 0) + 1

items = sorted(items, key=lambda i: i[0])
items = sorted(count.items(), key = lambda i: i[1], reverse = True)

# or equivalently:
# items = sorted(counts.items(), key = lambda i: (-1[i], i[0]))
#in this case, highest priority first
```

```
for user, count in counts.items():  
    print(f'{count:>7} {user}')
```