COS 316 Precept: Testing & Benchmarking

Overview

- What is testing?

- What is benchmarking?

Testing - Basic Approach in Go

- Source files and associated test files are placed in the same package/folder

- The name of the test file for any given source file is **`_test.go`**
  - E.g., **`router.go`** and **`router_test.go`**

- Import “**`testing`**”

- Test functions need to have the “**`Test`**” prefix, and the next character in the function name should be capitalized

Testing - Set up

```
> cd <COS316-Public repo>  # directory containing Vagrantfile>
#optional - start VM
> vagrant up ## if instance is not already started/provisioned
#optional - ssh to VM
> vagrant ssh
> git pull # update with precept4
```

Testing - Exercises

```
› cd precepts/precept4/mysort

# run test framework
› go test -v

# fix the bug and demonstrate tests pass
```

Benchmarking - Basic Approach in Go

- Benchmarks also reside in the **`_test.go`** files

- Import "**`testing`**"

- Test functions need to have the "**`Benchmark`**" prefix, and the next character in the function name should be capitalized

Benchmark - Exercises

- How to eliminate certain code in benchmarks?
  - **`b.ResetTimer(), b.StartTimer(), b.StopTimer()`**

- How to benchmark specific functions:
  - **`go test -bench=.  -benchtime=1s -run=Bench`**

- How to show memory allocations?
  - **`go test -bench=. -benchmem`**

    or
  - **`b.ReportAllocs()`**

Benchmark - Exercises

```
> cd precepts/precept4/fib

# run benchmark framework
> go test -bench=.

# will run for 10 seconds
> go test -bench=. -benchtime=10s

# will run experiment 10 times
> go test -bench=. -count=10
```

Testing and Benchmark - Exercises

```
› cd precepts/precept4/stack

# develop and run tests

# develop and run benchmarks
```

Questions:

1. **Does your testing framework pass all tests?**
2. **Does your benchmark(s) demonstrate improved performance?**