

Discussion 10

Testing

Kenneth Fang (kwf37), Newton Ni (cn279)

March 5, 2019

Agenda

1. Review Testing/Verification
 - ▶ Black Box vs. Glass Box
2. Rec 11 (please have it pulled up)
3. Test-Driven Development
 - ▶ Bounded Queue
 - ▶ List Set- No Dups

Review- Testing Strategies

- ▶ Black Box- test against an interface with no knowledge of the internal implementation

Review- Testing Strategies

- ▶ Black Box- test against an interface with no knowledge of the internal implementation
- ▶ Glass Box- test against interface with knowledge of the internal implementation

Review- Testing Strategies

- ▶ Black Box- test against an interface with no knowledge of the internal implementation
- ▶ Glass Box- test against interface with knowledge of the internal implementation
- ▶ Edge Cases- think about weird inputs that could potentially break your code

Review- Testing Strategies

- ▶ Black Box- test against an interface with no knowledge of the internal implementation
- ▶ Glass Box- test against interface with knowledge of the internal implementation
- ▶ Edge Cases- think about weird inputs that could potentially break your code
- ▶ Randomized Testing- generate random inputs to get more coverage with your tests (especially useful if you already implemented a `rep_ok` function).

Review- Testing Strategies

- ▶ Black Box- test against an interface with no knowledge of the internal implementation
- ▶ Glass Box- test against interface with knowledge of the internal implementation
- ▶ Edge Cases- think about weird inputs that could potentially break your code
- ▶ Randomized Testing- generate random inputs to get more coverage with your tests (especially useful if you already implemented a `rep_ok` function).
- ▶ Stress Testing- put in large inputs/datasets and see if your implementation breaks

Test-Driven Development

Why write tests before code?

Test-Driven Development

Why write tests before code?

- ▶ It helps you design your specs

Test-Driven Development

Why write tests before code?

- ▶ It helps you design your specs
- ▶ It allows you to redesign very quickly

Test-Driven Development

Why write tests before code?

- ▶ It helps you design your specs
- ▶ It allows you to redesign very quickly
- ▶ It gives your whole team a common understanding of the code

Bounded Queue

```
module type BoundedQueue = sig
  type 'a t
  val empty: int -> 'a t
  val enqueue: 'a -> 'a t -> 'a t
  val peek: 'a t -> 'a
  val dequeue: 'a t -> 'a t
end
```

List Set No Dups

```
module type ListSetNoDups = sig
  type 'a t
  val empty: 'a t
  val insert: 'a -> 'a t -> 'a t
  val remove: 'a -> 'a t -> 'a t
  val mem: 'a -> 'a t -> bool
end
```