

CS106B Midterm 1 Solutions

1. ADTs: Stacks and Queues

All function output:

```
{5, 4, 3, 1}           // provided
{5, 4, 3, 1, 1, 3}      // (a) Checkpoint 1
{5, 4, 3, 1}           // (b) Checkpoint 2
{5, 4, 3, 1, 3, 1}      // (c) Checkpoint 3
{1, 3, 4, 5}           // (d) Checkpoint 4
{1, 3, 4, 5}           // (e) Checkpoint 5
```

2. Big-O

- (a) $O(N)$
- (b) $O(N \log N)$
- (c) $O(N^2)$
- (d) $O(N)$
- (e) $O(N)$

3. C++ Strings and Testing Framework: Codebreaking

(a) These length 1 tests pass:

```
EXPECT_EQUAL(counts["A"], 1);
EXPECT_EQUAL(counts["Y"], 1);
```

(b) *All* tests of length 2 pass:

```
EXPECT_EQUAL(counts["AY"], 1);
EXPECT_EQUAL(counts["BO"], 2);
EXPECT_EQUAL(counts["OB"], 1);
EXPECT_EQUAL(counts["OO"], 1);
EXPECT_EQUAL(counts["SA"], 1);
EXPECT_EQUAL(counts["YS"], 1);
```

(c) These length 1 tests fail:

```
EXPECT_EQUAL(counts["B"], 3); // will incorrectly answer 2
EXPECT_EQUAL(counts["O"], 3); // will incorrectly answer 2
EXPECT_EQUAL(counts["S"], 2); // will incorrectly answer 1
```

(d) Since all length 2 tests pass, write N/A in both boxes.

4. ADTs: LLMs

(a)

```
void makeWordCounts(Vector<string>& tokens, Map<string, Map<string, int>>& counts)
{
    for (int i = 0; i < tokens.size() - 1; i++) { // note both < and - 1
        string word1 = tokens[i];
        string word2 = tokens[i + 1];
        counts[word1][word2]++;
    }
}
```

(b)

```
void makeLLM(Map<string, Map<string, int>>& counts, Map<string, Vector<Next>>& LLM)
{
    for (string first : counts) {
        double denom = 0.0; // by starting as double, don't need cast later
        for (string second : counts[first]) {
            denom += counts[first][second];
        }
        for (string second : counts[first]) {
            // must cast here if denominator was declared int
            LLM[first].add(Next(second, counts[first][second] / denom));
        }
    }
}
```