Project 4 Report

Notable Obstacles

Reading the spec

Stackoverflow

Test Data

• appendToAll

⁃ {"ed", "xavier", "", "eleni”}, -1, “+5” to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 4, “+5” to see if it appends correctly

• lookup

⁃ {"ed", "xavier", "", "eleni”}, -1, “eleni” to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 4, “xavier” to see if it finds it correctly

⁃ {"ed", "xavier", “xavier”, “xavier”}, -1, “xavier” to see if it handles multiple occurrences

⁃ {"ed", "xavier", "", "eleni”}, 4, “+5” to see if it handles no occurrences

• positionOfMax

⁃ {"ed", "xavier", "", "eleni”}, -1 to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 0 to see if it handles no max

⁃ {"ed", "xavier", "", "eleni”}, 4 to see if it finds the max correctly

⁃ {"ed", "xavier", “xavier”, "eleni”}, 4 to see if it finds the earliest max correctly

• rotateLeft

⁃ {"ed", "xavier", "", "eleni”}, -1, 0 to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 4, 5 to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 4, 0 to see if it rotates correctly

• countRuns

⁃ { "gavin", "gavin", "gavin", "xavier", "xavier" }, -1 to see if it handles bad input

⁃ { "gavin", "gavin", "gavin", "xavier", "xavier" }, 5 to see if it counts correctly

• flip

⁃ {"ed", "xavier", "", "eleni”}, -1, 0 to see if it handles bad input

⁃ {"ed", "xavier", "", "eleni”}, 4 to see if it flips correctly for even sizes

⁃ {"ed", "xavier", "", "eleni”, “bob”}, 5 to see if it flips correctly for odd sizes

• differ

⁃ {"betty", "john", "", "xavier", "kevin", "dianne”}, -1, {"betty", "john", "dianne", "", "xavier”}, 5 for bad input

⁃ {"betty", "john", "", "xavier", "kevin", "dianne”}, 6, {"betty", "john", "dianne", "", "xavier”}, -1 for bad input

⁃ {"betty", "john", "", "xavier", "kevin", "dianne”}, 6, {"betty", "john", "dianne", "", "xavier”}, 5 to test an intermediate break

⁃ {"betty", "john", "", "xavier", "kevin", "dianne”}, 6, {"betty", "john", "", "xavier", "kevin"}, 5 to test a break when one reaches end

⁃ {"betty", "john", "", "xavier", "kevin", "dianne”}, 6, {"betty", "john", "", "xavier", "kevin", "dianne”}, 6 to test a break when both reach end

• subsequence

⁃ {“a”, “b”, “a”, “b”, “a”}, -1, {“a”, “b”, “a”, “b”}, 4 for bad input

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “b”, “a”, “b”}, -1 for bad input

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “b”, “a”, “b”}, 4 to test one occurrence

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “b”, “a”}, 3 to test multiple occurrences

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “b”, “c”}, 4 to test zero occurrence

• lookupAny

⁃ {“a”, “b”, “a”, “b”, “a”}, -1, {“a”, “b”, “a”, “b”}, 4 for bad input

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “b”, “a”, “b”}, -1 for bad input

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“c”}, 1 for no matches

⁃ {“a”, “b”, “a”, “b”, “a”}, 5, {“a”, “c”}, 2 for multiple matches

⁃ {“a”, “b”, “c”, “d”, “e”}, 5, {“c”}, 1 for one match

• divide

⁃ {"dianne", "fiona", "gavin", "xavier", "ed", "betty”}, -1, “fiona” to test bad input

⁃ {"dianne", "fiona", "gavin", "xavier", "ed", "betty”}, 6, “A” to test divider that goes before everything

⁃ {"dianne", "fiona", "gavin", "xavier", "ed", "betty”}, 6, “z” to test divider that goes after everything

⁃ {"dianne", "fiona", "gavin", "xavier", "ed", "betty”}, 6, “f” to test divider that goes in between

⁃ {"dianne", "fiona", "gavin", "xavier", "ed", "betty”}, 6, “fiona” to test divider that goes before everything