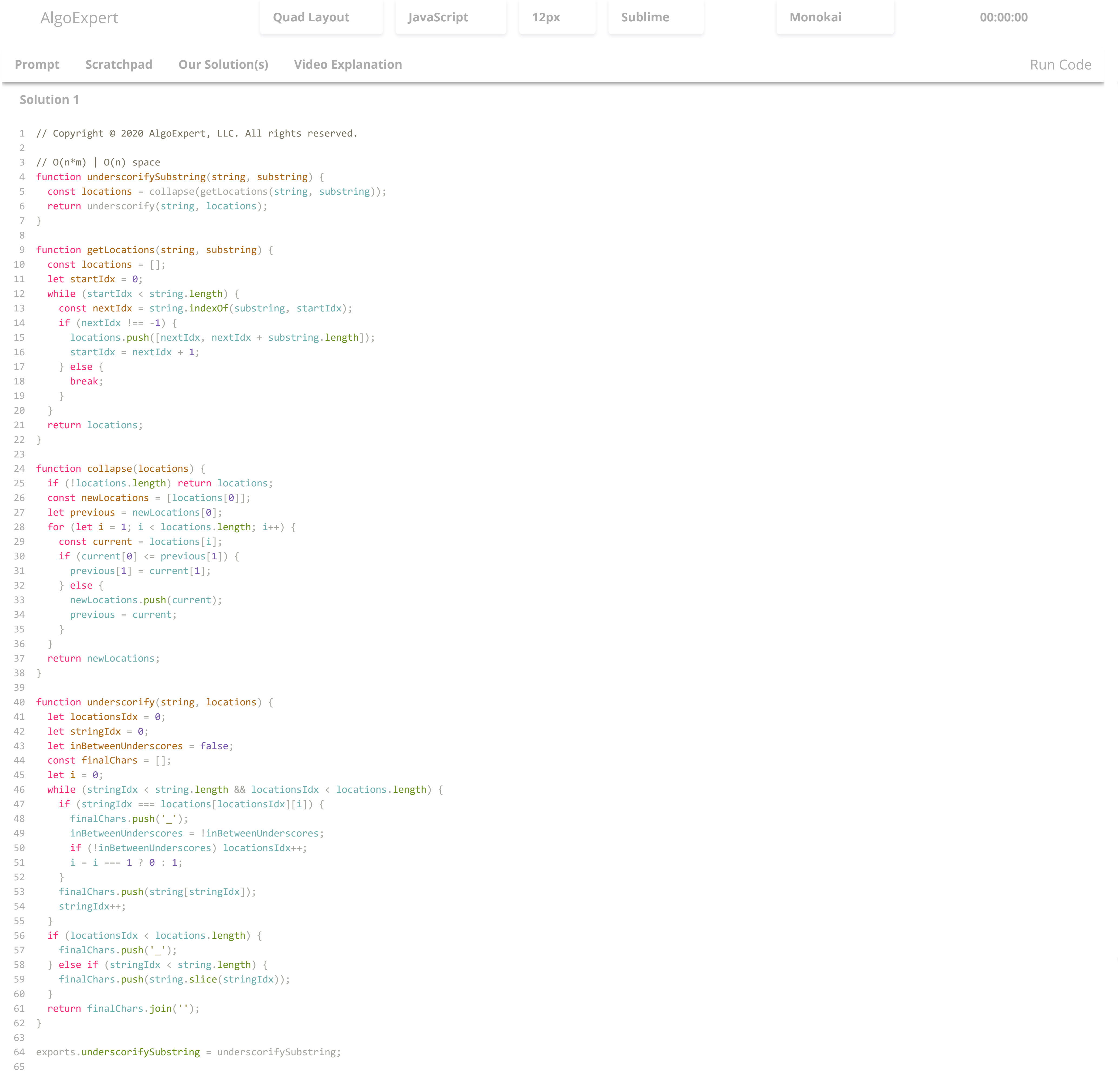

```javascript
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// O(n*m) | O(n) space
function underscorifySubstring(string, substring) {
  const locations = collapse(getLocations(string, substring));
  return underscorify(string, locations);
}

function getLocations(string, substring) {
  const locations = [];
  let startIdx = 0;
  while (startIdx < string.length) {
    const nextIdx = string.indexOf(substring, startIdx);
    if (nextIdx !== -1) {
      locations.push([nextIdx, nextIdx + substring.length]);
      startIdx = nextIdx + 1;
    } else {
      break;
    }
  }
  return locations;
}

function collapse(locations) {
  if (!locations.length) return locations;
  const newLocations = [locations[0]];
  let previous = newLocations[0];
  for (let i = 1; i < locations.length; i++) {
    const current = locations[i];
    if (current[0] <= previous[1]) {
      previous[1] = current[1];
    } else {
      newLocations.push(current);
      previous = current;
    }
  }
  return newLocations;
}

function underscorify(string, locations) {
  let locationsIdx = 0;
  let stringIdx = 0;
  let inBetweenUnderscores = false;
  const finalChars = [];
  let i = 0;
  while (stringIdx < string.length && locationsIdx < locations.length) {
    if (stringIdx === locations[locationsIdx][i]) {
      finalChars.push('_');
      inBetweenUnderscores = !inBetweenUnderscores;
      if (!inBetweenUnderscores) locationsIdx++;
      i = i === 1 ? 0 : 1;
    }
    finalChars.push(string[stringIdx]);
    stringIdx++;
  }
  if (locationsIdx < locations.length) {
    finalChars.push('_');
  } else if (stringIdx < string.length) {
    finalChars.push(string.slice(stringIdx));
  }
  return finalChars.join('');
}

exports.underscorifySubstring = underscorifySubstring;
```