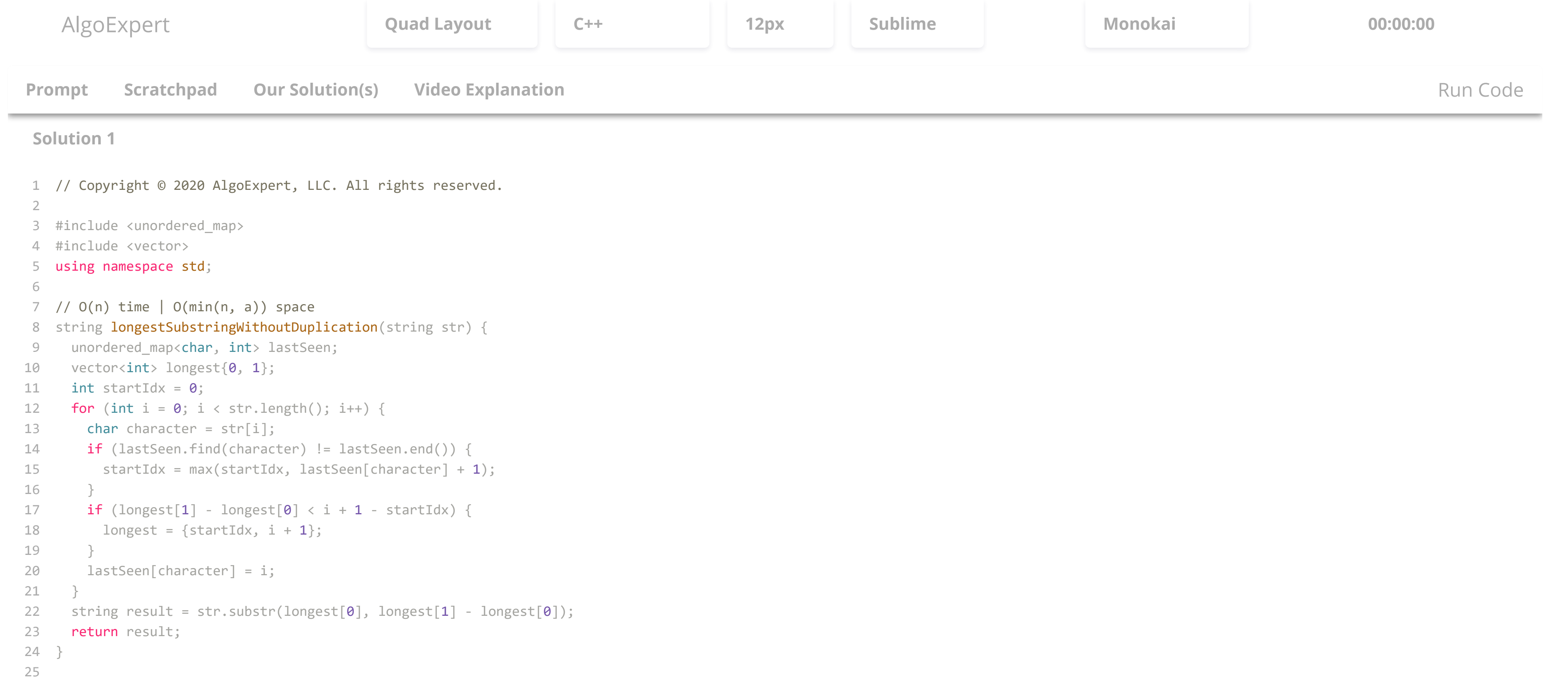

Solution 1

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

#include <unordered_map>
#include <vector>
using namespace std;

// O(n) time | O(min(n, a)) space
string longestSubstringWithoutDuplication(string str) {
  unordered_map<char, int> lastSeen;
  vector<int> longest{0, 1};
  int startIdx = 0;
  for (int i = 0; i < str.length(); i++) {
    char character = str[i];
    if (lastSeen.find(character) != lastSeen.end()) {
      startIdx = max(startIdx, lastSeen[character] + 1);
    }
    if (longest[1] - longest[0] < i + 1 - startIdx) {
      longest = {startIdx, i + 1};
    }
    lastSeen[character] = i;
  }
  string result = str.substr(longest[0], longest[1] - longest[0]);
  return result;
}

```