Our Solution(s) Run Code

Solution 1

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

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

// O(n) time | O(n) space
bool balancedBrackets(string str) {
  string openingBrackets = "([{";
  string closingBrackets = ")]}";
  unordered_map<char, char> matchingBrackets{
      {')', '('}, {']', '['}, {'}', '{'}};
  vector<char> stack;
  for (char character : str) {
    if (openingBrackets.find(character) != string::npos) {
      stack.push_back(character);
    } else if (closingBrackets.find(character) != string::npos) {
      if (stack.size() == 0) {
        return false;
      }
      if (stack[stack.size() - 1] == matchingBrackets[character]) {
        stack.pop_back();
      } else {
        return false;
      }
    }
  }
  return stack.size() == 0;
}
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
using namespace std;

bool balancedBrackets(string str) {
  // Write your code here.
  return false;
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.