

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <climits>
4 #include <unordered_map>
5 #include <vector>
6
7 using namespace std;
8
9 string smallestSubstringContaining(string bigString, string smallStrin
10 unordered_map<char, int> getCharCounts(string str);
11 vector<int> getSubstringBounds(string str,
12                               unordered_map<char, int> targetCharCou
13 vector<int> getCloserBounds(int idx1, int idx2, int idx3, int idx4);
14 string getStringFromBounds(string str, vector<int> bounds);
15 void increaseCharCount(char c, unordered_map<char, int> &charCounts);
16 void decreaseCharCount(char c, unordered_map<char, int> &charCounts);
17
18 // O(b + s) time | O(b + s) space - where b is the length of the big
19 // input string and s is the length of the small input string
20 string smallestSubstringContaining(string bigString, string smallStrin
21     unordered_map<char, int> targetCharCounts = getCharCounts(smallStrin
22     vector<int> substringBounds = getSubstringBounds(bigString, targetCh
23     return getStringFromBounds(bigString, substringBounds);
24 }
25
26 unordered_map<char, int> getCharCounts(string str) {
27     unordered_map<char, int> charCounts;
28     for (auto c : str) {
29         increaseCharCount(c, charCounts);
30     }
31     return charCounts;
32 }
33
```

Solution 1

Solution 2

Solution 3

```
1 using namespace std;
2
3 string smallestSubstringContaining(string bigString, string smallStrin
4     // Write your code here.
5     return "";
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <climits>
4 #include <unordered_map>
5 #include <vector>
6
7 using namespace std;
8
9 string smallestSubstringContaining(string bigString, string smallStrin
10 unordered_map<char, int> getCharCounts(string str);
11 vector<int> getSubstringBounds(string str,
12                               unordered_map<char, int> targetCharCou
13 vector<int> getCloserBounds(int idx1, int idx2, int idx3, int idx4);
14 string getStringFromBounds(string str, vector<int> bounds);
15 void increaseCharCount(char c, unordered_map<char, int> &charCounts);
16 void decreaseCharCount(char c, unordered_map<char, int> &charCounts);
17
18 // O(b + s) time | O(b + s) space - where b is the length of the big
19 // input string and s is the length of the small input string
20 string smallestSubstringContaining(string bigString, string smallStrin
21     unordered_map<char, int> targetCharCounts = getCharCounts(smallStrin
22     vector<int> substringBounds = getSubstringBounds(bigString, targetCh
23     return getStringFromBounds(bigString, substringBounds);
24 }
25
26 unordered_map<char, int> getCharCounts(string str) {
27     unordered_map<char, int> charCounts;
28     for (auto c : str) {
29         increaseCharCount(c, charCounts);
30     }
31     return charCounts;
32 }
33
```

```
1 using namespace std;
2
3 string smallestSubstringContaining(string bigString, string smallStrin
4     // Write your code here.
5     return "";
6 }
7
```

