4 Handwritten Problem

This problem is to be submitted independently. We recommend trying it on your own, checking your answer with a group and discussing solutions, and then submitting it to Gradescope. These will be graded on completion, not by correctness. However, we want to see that you were thinking about the problem. Please implement your solution in the space below, in `anagram.h`, or on a piece of paper. The starter files can be found on Canvas.

17. Write a function that takes in two strings and returns whether they are anagrams of each other (words that contain the same letters). The only characters will be spaces and lowercase letters. Do this in $\Theta(n)$ time.

- Example 1: Given s1 = "anagram" and s2 = "nagaram", return true.
- Example 2: Given s1 = "i love eecs" and s2 = "i scole ve e", return true.
- Example 3: Given s1 = "anagrams" and s2 = "anagrams anagrams", return false.
- Example 4: Given s1 = "cats" and s2 = "cat", return false.

```
// check if two strings are anagrams
bool isAnagram(const string &s1, const string &s2);
```

```
for (size_t i = 0; i < s1.length(); i++) {
        if (s1.at(i) == ' ') {
                s1.erase(i, 1);
        }
}
for (size_t i = 0; i < s2.length(); i++) {
        if (s2.at(i) == ' ') {
                s2.erase(i, 1);
        }
}

if (s1.length() == s2.length()) {
    while (!s1.empty() && !s2.empty()) {
        if( s2.find(s1.at(0)) != string::npos) {
            s2.erase(s2.find(s1.at(0)), 1);
            s1.erase(0, 1);
        }
        else { return false; }
    }

    return true;
}
else {
  return false;
}
```