```cpp
class Solution {
public:
    string frequencySort(string s) {
        string sorted;
        int freq;
        // Vector to store pairs of character and frequency
        vector<pair<char, int>> charFreq;

        for (int i = 0; i < s.length(); i++) {
            // Check if the character is already in the vector
            //'auto it' stores the iterator returned by find_if
            // determines if a character is already in the charFreq vector.
            auto it = find_if(
                charFreq.begin(), charFreq.end(),
                [&](const pair<char, int>& p) { return p.first == s[i]; });

            // If the character is not found, cal the freq and add to vector
            if (it == charFreq.end()) {
                freq = count(s.begin(), s.end(), s[i]);
                charFreq.push_back({s[i], freq});
            }
        }

        // Sort the vector by freq in descending order using lambda function
        sort(charFreq.begin(), charFreq.end(),
             [](const pair<char, int>& a, const pair<char, int>& b) {
                 // Compare by frequency in descending order
                 return a.second > b.second;
             });

        // Append vector into String
        for (const auto& p : charFreq) {
            // Append character 'p.first' 'p.second' times
            // This makes sure that each char appearance = frequency
            sorted.append(p.second, p.first);
        }
        return sorted;
    }
};
```