Valid Anagram

```
class Solution {
public:
    bool isAnagram(string s, string t) {
      int i;

      unordered_map<char,int> maps;

      unordered_map<char,int> mapt;

      if (s.size() != t.size()) return false;

      for (i = 0;i<s.size();i++){
        maps[s[i]] += 1;
        mapt[t[i]] += 1;
      }

      if (maps.size() != mapt.size()) return false;

      for (auto j: maps){
        if (mapt[j.first] != j.second) return false;
      }

      return true;
    }
};
```

- Working solution
  - if strings aren't equal length, they cant be valid anagrams
  - created hash tables of frequency of each letter in the strings
  - if hash tables aren't equal size, they cant be valid anagrams

- iterate through hash tables and compare values, if any frequency values are not equal for the same letter key, cant be valid anagram