

# Group Anagram

```
class Solution {
public:
    vector<vector<string>> groupAnagrams(vector<string>& strs) {
        int i;
        vector<vector<string>> awn;
        unordered_map<string, vector<string>> map;
        string curr;

        for (i = 0; i<strs.size();i++){
            curr = strs[i];
            sort(curr.begin(),curr.end());
            map[curr].push_back(strs[i]);
        }

        for (auto j: map){
            awn.push_back(j.second);
        }

        return awn;
    }
};
```

- working solution
  - two words are anagrams if they are the same length and when sorted are the same
  - sort every string and use the sorted value as a key then add the original to the value which is a list of values that are the same when sorted (AKA anagrams)