

week07–习题

202.实现Trie（前缀树）

```
1 class Trie {
2     public:
3         Trie() {}
4         ~Trie() {
5             for (int i = 0; i < 26; i++) {
6                 if (next[i] == nullptr) continue;
7                 delete(next[i]);
8                 next[i] = nullptr;
9             }
10        }
11        void insert(string word) {
12            Trie *node = this;
13            for (auto c : word) {
14                if (node->next[c - 'a'] == nullptr) {
15                    node->next[c - 'a'] = new Trie();
16                }
17                node = node->next[c - 'a'];
18            }
19            node->is_string = true;
20        }
21        bool search(string word) {
22            Trie *node = this;
23            for (auto c : word) {
24                if (node->next[c - 'a'] == nullptr) return false;
25                node = node->next[c - 'a'];
26            }
27            return node->is_string;
28        }
29        bool startsWith(string prefix) {
30            Trie *node = this;
31            for (auto c : prefix) {
32                if (node->next[c - 'a'] == nullptr) return false;
33                node = node->next[c - 'a'];
34            }
35            return true;
36        }
37    }
```

```

43
44 private:
45     bool is_string = false;
46     Trie* next[26] = {nullptr};
47 };
48

```

36.有效的数独

```

1  class Solution {
2  public:
3      bool isValidSudoku(vector<vector<char>>& board) {
4          int row[9][10] = {0}; // 哈希表存储每一行的每个数是否出现过，默认初始情况
           下，每一行每一个数都没有出现过
5          // 整个board有9行，第二维的维数10是为了让下标有9，和数独中的数字9对应。
6          int col[9][10] = {0}; // 存储每一列的每个数是否出现过，默认初始情况下，每
           一列的每一个数都没有出现过
7          int box[9][10] = {0}; // 存储每一个box的每个数是否出现过，默认初始情况
           下，在每个box中，每个数都没有出现过。整个board有9个box。
8          for(int i=0; i<9; i++){
9              for(int j = 0; j<9; j++){
10                 // 遍历到第i行第j列的那个数，我们要判断这个数在其所在的行有没有出
           现过，
11                 // 同时判断这个数在其所在的列有没有出现过
12                 // 同时判断这个数在其所在的box中有没有出现过
13                 if(board[i][j] == '.') continue;
14                 int curNumber = board[i][j] - '0';
15                 if(row[i][curNumber]) return false;
16                 if(col[j][curNumber]) return false;
17                 if(box[j/3 + (i/3)*3][curNumber]) return false;
18                 row[i][curNumber] = 1; // 之前都没出现过，现在出现了，就给它置为
           1，下次再遇见就能够直接返回false了。
21                 col[j][curNumber] = 1;
22                 box[j/3 + (i/3)*3][curNumber] = 1;
23             }
24         }
25         return true;
26     }
27 };
28

```