Description

71. Simplify Path

Given an absolute path for a file (Unix-style), simplify it.

```
For example,
path = "/home/", => "/home"
path = "/a/./b/../../c/", => "/c"
```

Idea

This is a pure constructing problem, we can split the whole input string by "/", then we add each splitted string into a container. Everytime we see an empty string or a ".", we just skip it. If we see ".." we need to remove the last string that we just added to the container, so this is last in first out data structure, so we use a stack with O(n) time complexity.

Java

```
class Solution {
    public String simplifyPath(String path) {
        String[] strs = path.split("/");
        Set<String> set = new HashSet<>(Arrays.asList("", ".", ".."));
        Deque<String> stack = new ArrayDeque<>();
        for (String str : strs) {
            if (str.equals("..") && !stack.isEmpty()) {
                stack.pollLast();
            } else if (!set.contains(str)){
                stack.offerLast(str);
            }
        }

        String res = String.join("/", stack);
        return "/" + res;
    }
}
```

For C++, we need to write our own splitter.

C++

```
class Solution {
public:
    string simplifyPath(string path) {
        vector<string> paths;
        split(path, '/', paths);
        unordered_set<string> set = {"", ".", ".."};
        vector<string> strs;

        for (int i = 0; i < paths.size(); i++) {
            if (paths[i] == ".." && !strs.empty()) {
                strs.pop_back();
            } else if (set.find(paths[i]) == set.end()) {
                strs.push_back(paths[i]);
            }
        }

        string res;
        for (auto str : strs) {
            res += "/" + str;
        }
        return res.empty() ? "/" : res;
    }

    void split(string str, char delimiter, vector<string>& vec) {
        stringstream ss(str);
        string temp;
        while (getline(ss, temp, delimiter)) {
            vec.push_back(temp);
        }
    }
};
```

Summary

- Stack
- Tried the "basic calculator" template, some of the corner cases are hard to handle.