1. 题目
2. 题干

以 Unix 风格给出一个文件的绝对路径，你需要简化它。或者换句话说，将其转换为规范路径。

在 Unix 风格的文件系统中，一个点（.）表示当前目录本身；此外，两个点 （..） 表示将目录切换到上一级（指向父目录）；两者都可以是复杂相对路径的组成部分。

请注意，返回的规范路径必须始终以斜杠 / 开头，并且两个目录名之间必须只有一个斜杠 /。最后一个目录名（如果存在）不能以 / 结尾。此外，规范路径必须是表示绝对路径的最短字符串。

1. 示例

示例 1：

输入："/home/"

输出："/home"

解释：注意，最后一个目录名后面没有斜杠。

示例 2：

输入："/../"

输出："/"

解释：从根目录向上一级是不可行的，因为根是你可以到达的最高级。

示例 3：

输入："/home//foo/"

输出："/home/foo"

解释：在规范路径中，多个连续斜杠需要用一个斜杠替换。

示例 4：

输入："/a/./b/../../c/"

输出："/c"

示例 5：

输入："/a/../../b/../c//.//"

输出："/c"

示例 6：

输入："/a//b////c/d//././/.."

输出："/a/b/c"

1. 题解
2. 思路

题目看完一遍，毫无头绪，简直就是阅读理解题。再细细思考，可以看出这是一个化简字符串的题目，要求是将不合规范的路径修改为合规的路径，且路径的长度最短，即需要将不符合规范的字符给删掉，将多余的字符也删掉。如遇到”.”，表示当前目录，那么直接删掉即可；若是”..”，表示上级目录，那么就需要返回到上一级目录中；若是有多个”/”，则只取一个即可。

因此，首先我们需要将字符串给拆分开，以“/”字符为分隔符，这样生成了一个String数组，根据数组中的每个String来进行操作。这里，我们使用一个栈stack来存储真正的路径，若数组中有”..”，则表示需要回到上一级目录中，则弹出栈的顶部；若是“.”，则表示当前目录，不操作；若是“”，空字符串，也是不操作；若是其他的字符，就是真实的地址，需要入栈。最后，将栈中的字符一一输出即可。

1. 代码实现

Java：

class Solution {

    public String simplifyPath(String path) {

        String[] s = path.split("/");

        Stack<String> stack = new Stack<>();

        for(int i = 0 ; i < s.length; i++){

            if(!stack.isEmpty() && s[i].equals("..")){

                stack.pop();

            }else if(!s[i].equals(".") && !s[i].equals("") && !s[i].equals("..")){

                stack.push(s[i]);

            }

        }

        if(stack.isEmpty()){

            return "/";

        }

        StringBuilder out = new StringBuilder();

        for(int i = 0; i < stack.size(); ++i){

            out.append("/" + stack.get(i));

        }

        return out.toString();

    }

}