# [简化路径](https://leetcode-cn.com/explore/interview/card/bytedance/242/string/1013/)

**头条重点**

## 题目

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。

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"

## 解题思路

1. 利用栈的特性，将有效路径名压入
2. 当遇到 .. 时 pop 栈
3. 最后按顺序 pop 组成最终路径

public static String simplifyPath(String path) {
 ArrayDeque<String> stack = new ArrayDeque<>();
 String[] split = path.split("/");
 for (String s : split) {
 if (s.isEmpty()) {
 continue;
 }

 switch (s) {
 case "..":
 stack.pollLast();
 break;
 case ".":
 break;
 default:
 stack.offerLast(s);
 }
 }

 StringBuilder builder = new StringBuilder("/");
 for (String s : stack) {
 builder.append(s);
 builder.append("/");
 }

 if (builder.length() > 1) {
 builder.deleteCharAt(builder.length() - 1);
 }
 return builder.toString();
}