71. Simplify Path

Given an absolute path for a Unix-style file system, which begins with a slash '/', transform this path into its simplified canonical path.

In Unix-style file system context, a single period '.' signifies the current directory, a double period ".." denotes moving up one directory level, and multiple slashes such as "//" are interpreted as a single slash. In this problem, treat sequences of periods not covered by the previous rules (like "...") as valid names for files or directories.

The simplified canonical path should adhere to the following rules:

It must start with a single slash '/'.

Directories within the path should be separated by only one slash '/'.

It should not end with a slash '/', unless it's the root directory.

It should exclude any single or double periods used to denote current or parent directories.

Return the new path.

Example 1:

- **Input:** path = "/home/"
- **Output:** "/home"
- **Explanation:** The trailing slash should be removed.

Example 2:

- **Input:** path = "/home//foo/"
- **Output:** "/home/foo"
- **Explanation:** Multiple consecutive slashes are replaced by a single one.

Example 3:

- **Input:** path = "/home/user/Documents/../Pictures"
- **Output:** "/home/user/Pictures"
- **Explanation:** A double period ".." refers to the directory up a level.

Example 4:

- **Input:** path = "/../"
- **Output:** "/"
- **Explanation:** Going one level up from the root directory is not possible.

Example 5:

- **Input:** path = "/.../a/../b/c/../d/./"
- **Output:** "/.../b/d"
- **Explanation:** "..." is a valid name for a directory in this problem.

Constraints:

- 1 <= path.length <= 3000
- path consists of English letters, digits, period '.', slash '/' or '_'.
- path is a valid absolute Unix path.