Solution 1

```
1   // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3   using System;
4   using System.Linq;
5   using System.Collections.Generic;
6
7   public class Program {
8     // O(n) time | O(n) space - where n is the length of the path
9     public static string ShortenPath(string path) {
10      bool startsWithPath = path[0] == '/';
11      string[] tokensArr = path.Split("/");
12      List<string> tokensList = new List<string>(tokensArr);
13      List<string> filteredTokens = tokensList.FindAll(token => isImportantToken(token));
14      Stack<string> stack = new Stack<string>();
15      if (startsWithPath) stack.Push("");
16      foreach (string token in filteredTokens) {
17        if (token.Equals("..")) {
18          if (stack.Count == 0 || stack.Peek().Equals("..")) {
19            stack.Push(token);
20          } else if (!stack.Peek().Equals("")) {
21            stack.Pop();
22          }
23        } else {
24          stack.Push(token);
25        }
26      }
27
28      if (stack.Count == 1 && stack.Peek().Equals("")) return "/";
29      var arr = stack.ToArray();
30      Array.Reverse(arr);
31      return String.Join("/", arr);
32    }
33
34    public static bool isImportantToken(string token) {
35      return token.Length > 0 && !token.Equals(".");
36    }
37  }
38
```