Solution 1

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

import "strings"

// O(n) time | O(n) space - where n is the length of the path
func ShortenPath(path string) string {
  startsWithSlash := path[0] == '/'
  splits := strings.Split(path, "/")
  tokens := []string{}
  for _, token := range splits {
    if isImportantToken(token) {
      tokens = append(tokens, token)
    }
  }

  stack := []string{}
  if startsWithSlash {
    stack = append(stack, "")
  }
  for _, token := range tokens {
    if token == ".." {
      if len(stack) == 0 || stack[len(stack)-1] == ".." {
        stack = append(stack, token)
      } else if stack[len(stack)-1] != "" {
        stack = stack[:len(stack)-1]
      }
    } else {
      stack = append(stack, token)
    }
  }

  if len(stack) == 1 && stack[0] == "" {
    return "/"
  }
  return strings.Join(stack, "/")
}

func isImportantToken(token string) bool {
  return len(token) > 0 && token != "."
}

```