Programming Problem - Search - Part 2

Instructions

You must have solved Part 1 of this problem and submitted it for review prior to your in-office interview/coding round.

You will be expected to solve Part 2 of the problem in our office as part of the next phase of our interview process.

You will be given a laptop with Ruby installed and source code of solution that you have submitted for Part 1.

Problem Statement

Part 1 of this problem mentioned the following -

Additional Considerations
Although the search strength algorithm described here is quite simple, developers should make provision for substituting a more complex method in the future and consider the impact of nested pages.

Problem 1:

It's high time we enhance our search solution to be able to handle nested pages

A Page can have nested pages e.g. P1 can have nested pages P1.1, P1.2, P1.3 etc
When a Query matches a given nested page, it's score is calculated as per the algorithm in Part 1. However, since this nested page belongs to a parent page, we should add 10% of the nested page's score to the parent page's score.

For example, assume the following web pages and keyword lists:
Page 1: Ford, Car, Review
Page 1.1: Ford, Review
Page 2: Toyota, Car
Page 2.1: Ford
Page 3: Car, Ford

For N equal 8, a query with keywords Ford and Car in that order yields the following strength ratings.
Page 1.1: (8x8) = 64
Page 1: (8x8 + 7x7 + 64x0.1) = 119.4

Page 2.1: (8x8) = 64
Page 2: (7x7 + 64x0.1) = 55.4
Page 3: (7x8) = 56.

Display only the parent pages in the search results e.g. if P1 does not match but P1.1 matches a query, show only P1 and not P1.1 in the results output

Input

Code letters P denote a parent page and PP denote a child page under the previous parent. All PPs denote children/nested pages under the most recent P. A new P denotes a new parent page.

Sample Input
P Ford Car
PP Review Car
PP Review Ford
P Toyota Car
PP Car

Q Ford
Q Car
Q Review

Output for the Sample Input

Q1: P1
Q2: P2 P1
Q3: P1

End of problem