



Search Engine

Link submit: <https://www.hackerearth.com/practice/data-structures/advanced-data-structures/trie-keyword-tree/practice-problems/algorithm/search-engine/>

Solution:

C++	http://ideone.com/0XJniF
Java	https://ideone.com/Cw9hsl
Python	https://ideone.com/3n36A3

Tóm tắt đề:

Cho bạn danh sách các từ và độ ưu tiên của nó. Cho bạn q câu truy vấn. Hãy liệt kê ra độ ưu tiên của từ lớn nhất (liệt kê ra một con số).

Input:

Dòng đầu tiên chứa hai số N và q lần lượt là số lượng từ và số lượng câu truy vấn.

n dòng tiếp theo, mỗi dòng gồm một chuỗi s và một số nguyên $weight$ lần lượt là từ trong danh sách và độ ưu tiên của nó.

q dòng tiếp theo, mỗi dòng chứa mỗi chuỗi t là từ người dùng tìm kiếm.

($1 \leq n, weight, |s|, |t| \leq 10^6, 1 \leq q \leq 10^5$, với $|s|$ là độ dài của chuỗi s)

Output:

In ra một số duy nhất là từ có độ ưu tiên cao nhất khi tìm kiếm.

Ví dụ:

2 1 hackerearth 10 hackerrank 9 hacker	10
-------------------------------------------------	----

Hướng dẫn giải:

Lần lượt để các từ trong danh sách vào hàng đợi ưu tiên, từ nào có độ ưu tiên cao hơn sẽ nằm trong danh sách trước.

Cây Trie được thêm vào tham số mới với int $weight$ (độ ưu tiên của từ).

Sau khi để xong vào hàng đợi ưu tiên, chúng ta sẽ lấy từng từ ra và bỏ vào trong cây Trie (việc để vào hàng đợi ưu tiên giúp những từ nào có độ ưu tiên cao sẽ được thêm vào cây Trie trước).

Khi bỏ vào Trie xong, mỗi truy vấn bạn sẽ đi xét, nếu từ đó được tìm đến ký tự cuối cùng, và nếu ký tự cuối cùng này là tiền tố của một từ nào đó khác thì sẽ lấy độ ưu tiên của từ khác đó in ra và đây chính là kết quả bài toán.

Độ phức tạp: $O(\text{string_length} * (n + q))$, trong đó, thao tác thêm từ vào cây $O(\text{string_length} * n)$ với n là số lượng từ và thao tác tìm từ có độ ưu tiên cao nhất $O(\text{string_length} * q)$ với q là số lượng truy vấn.

Big-O Coding