



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/lohiDi

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

Input

Dòng đầu tiên chứa 2 số N và q lần lượt là số lượng từ có trong danh sách và số lượng câu truy vấn ($1 \leq n, \text{weight}, |s|, |t| \leq 10^6, 1 \leq q \leq 10^5$, với |s| là độ dài của chuỗi s).

n dòng tiếp theo lần lượt chứa các từ trong danh sách và độ ưu tiên của nó.

q dòng tiếp theo là chứa từ người dùng tìm kiếm.

Output

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

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

Hướng dẫn giải:

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

Khi bỏ vào xong, mỗi lần gặp truy vấn q bạn sẽ lấy từ đó có ra xét, từ đó được tìm đến ký tự cuối cùng, và nếu ký tự cuối cùng này là tiền tố của 1 từ nào đó khác thì sẽ lấy độ ưu tiên của khác đó in ra và đây chính là kết quả bài toán.

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

Big-O Coding