**wordgame:**

Vì xâu tạo thành phải là 1 xâu đối xứng nên ta sẽ xây xâu theo chiến thuật: xây từ 2 phía trái, phải một cách **đồng thời**. Xây từ phía trái là xây tiền tố, xây từ phía phải là xây hậu tố.

Gọi A, B thứ tự là tiền tố, hậu tố **hiện tại** của xâu cần dựng. Đồng thời, duy trì 2 biến L, R. L là độ dài tiền tố dài nhất của S mà tiền tố đấy có thể nhận được bằng cách xóa đi một số ký tự của A. R là độ dài hậu tố dài nhất của S mà hậu tố đấy có thể nhận được bằng cách xóa đi một số ký tự của B. Ban đầu, A = B = “”, L = R = 0.

Xâu cần dựng nhận được bằng cách ghép A với B => xâu cần dựng chứa từ khóa S khi và chỉ khi L + R >= độ dài S.

Giả sử độ dài A > độ dài B.

Ta sẽ tiếp tục xây về phía phải, tức là mở rộng B => cần tìm một từ w để ghép vào đầu xâu B (tức là B mới = w + B cũ).

Ví dụ A = “abcde”, B = “ba”. Do tính chất đối xứng, nên từ w phải có dạng: “c”, “dc”, “edc” hoặc “??...??edc” => A thừa thông tin, 2 ký tự đầu “ab” của A không được sử dụng đến => để biểu diễn A ta chỉ cần lưu thông tin hậu tố của A là “cde” là đủ. Đồng thời, để biết w, ta cũng ko cần thông tin xâu B => ko cần biểu diễn B trong TH này.

Nếu độ dài A < độ dài B, ta sẽ tiếp tục mở rộng A. Tương tự, ko cần lưu A trong TH này và chỉ cần biểu diễn B bằng tiền tố của B là đủ.

Trạng thái của ta sẽ là (id, type, remain, L, R).

Trong đó:

* type = 0 hoặc 1 tương ứng là độ dài A > B hay độ dài A < B.
* id nhận giá trị từ 1 tới N. Nếu độ dài A > độ dài B, từ có chỉ số id là hậu tố của A. Nếu độ dài A < B, từ có chỉ số id là tiền tố của B.
* remain = | độ dài A - độ dài B |
* L, R ý nghĩa như đã nói.

Qhđ trên tập trái thái trên.