

# HACKER

Ta sẽ sử dụng một cây IT để lưu trữ mã Hash của các xâu con của mật khẩu mới. Ta cũng sẽ phải lưu lại mã Hash của các xâu con của mật khẩu cũ bằng cách tính tổng tiền tố.

Để thực hiện truy vấn thứ hai, ta có thể sử dụng kĩ thuật lazy update. Thông tin mà cần lưu trữ ở các nút "lazy" là xâu con của mật khẩu cũ (bao gồm mã Hash và vị trí của nó) mà ta chuẩn bị thay thế với xâu con tương ứng trong mật khẩu mới. Để chuyển thông tin của nút "lazy" sang các nút con của nó, ta cần tìm mã Hash của hai đoạn tương ứng với hai nút con của mật khẩu cũ bằng cách sử dụng tổng tiền tố mà ta đã tạo ở trên.

Để thực hiện truy vấn thứ ba, xét xâu  $w = w_{n-1}...w_1w_0$  và mã Hash  $w_{n-1}.p^{n-1} + ... + w_1.p + w_0$  của nó. Ta có thể nhóm mã Hash theo  $w_i$  như sau:  $1.(p^{a_1} + ... + p^{a_{n_1}}) + 2.(p^{b_1} + ... + p^{b_{n_2}}) + ... + 26.(p^{z_1} + ... + p^{z_{n_{26}}})$  và đặt  $f_1 = p^{a_1} + ... + p^{a_{n_1}}$ ;  $f_2 = p^{b_1} + ... + p^{b_{n_2}}$ ; ...;  $f_{26} = p^{z_1} + ... + p^{z_{n_{26}}}$ . Khi đó, để thực hiện truy vấn thứ ba, ta chỉ cần đảo lại các  $f$ , tức là  $f'_1 = f_{26}, f'_2 = f_1, ...$ . Để có thể sử dụng lazy update cho truy vấn này, ta cần lưu thêm phần dư khi chia số lần thực hiện truy vấn 3 với 26.

Độ phức tạp:  $O(N.A + Q.\log N.A)$  với  $A$  là số lượng chữ cái trong bảng chữ cái.