* Ta sẽ tạo 1 mảng f[i][j][u][v] mang ý nghĩa là độ dài hàng tương đồng bậc k dài nhất tính từ vị trí (i,j) của bảng thứ nhất và vị trí (u,v) của bảng thứ 2. Dễ có f[i][j][u][v]=f[i][j+1][u][v+1] + 1 nếu Ai,j tương đồng với Bu,v và bằng 0 nếu Ai,j­ không tương đồng với Bu,v.
* Ta sẽ xét lần lượt: nếu cố định ô (i,j) cho so sánh với ô (u,v), từ đó ta có 1 dãy các đoạn từ (i+1,j), (i+2,j),…(i+k,j) so sánh với (u+1,v), (u+2,v) …(u+k,v). Ta có thể chọn hình chữ nhật có độ dài 1 cạnh là 2 đầu (x,y) bất kì trong đoạn từ 0 đến k và cạnh còn lại là min ( f[i+x][j][u+x][v], f[i+x+1][j][u+x+1][v], … f[i+y][j][u+y][v] ).Ta cần chọn hình chữ nhật có diện tích lớn nhất trong các hình chữ nhật này.
* Từ đây, ta sẽ xét 1 bài toán con sau: cho 1 dãy gồm h1,h2,…hk phần tử nguyên dương, giá trị của đoạn (i,j) là min(hi,hi+1,…hj)\*(j-i+1). Tìm giá trị lớn nhất của dãy. Ta sẽ giải bài toán trên trong O(n) bằng cách dùng cấu trúc dữ liệu kiểu Stack: với mỗi giá trị hi, ta kiểm tra xem đoạn dài nhất nào chứa có giá trị hi về cả 2 bên trái và phải, từ đó có thể lấy max và in ra kết quả.
* Độ phức tạp : O(n4), bộ nhớ sử dụng: O(n4).