

Bài yêu cầu tìm xâu con chung dài nhất của 2 xâu

Ví dụ:

AEYABC GWAK
AEUTUEBC

→ Xâu con chung dài nhất gồm 4 kí tự AEBC

Sử dụng thuật toán quy hoạch động

```
#include<iostream>
#include<algorithm>
using namespace std;
int main () {
    // Nhập hai chuỗi
    string s1, s2;
    cin >> s1 >> s2;

    // Lấy độ dài của các chuỗi
    int len1 = s1.length();
    int len2 = s2.length();

    // Khởi tạo mảng dp với tất cả các phần tử là 0
    int dp [len2] = {0};

    // Duyệt qua tất cả các phần tử trong chuỗi thứ nhất
    for (int i = 0 ; i < len1 ; i++) {
        int max_len = 0;
        // Duyệt qua tất cả các phần tử trong chuỗi thứ hai
        for (int j = 0 ; j < len2 ; j++) {
            // Nếu phần tử thứ i của chuỗi thứ nhất bằng phần tử thứ j của chuỗi thứ hai
            if (s2[j] == s1[i] && max_len + 1 != dp[j]) {
                // Cập nhật độ dài lớn nhất của xâu con chung dài nhất kết thúc bằng s1[i]
                dp[j] = max_len + 1;
            }
            else
                // Cập nhật độ dài lớn nhất của xâu con chung dài nhất trước vị trí j
                max_len = max(max_len, dp[j]);
        }
    }
    /*
    Xét các phần tử trong chuỗi thứ hai. Nếu phần tử s1[i] = s2[j] thì xâu con tăng dài nhất kết thúc bằng s1[i]
    tính đến vị trí j có độ dài lớn nhất =
    độ dài lớn nhất của xâu con chung dài nhất trước vị trí j cộng thêm 1;
    */
}
```

```

max_len+1 khác dp[j] tức là phần tử đó đã có trước đó để hàm else chạy để thay đổi max_len tăng lên 1.
*/

// In ra độ dài lớn nhất của xâu con chung dài nhất
cout << *max_element(dp, dp + len2);
}

```

Ví dụ:

Nhập vào:

s1 = "abcde"

s2 = "ace"

Kết quả xuất ra:

Mảng dp sau khi duyệt qua phần tử 0 của chuỗi thứ nhất: 1 0 0

Mảng dp sau khi duyệt qua phần tử 1 của chuỗi thứ nhất: 1 0 0

Mảng dp sau khi duyệt qua phần tử 2 của chuỗi thứ nhất: 1 2 0

Mảng dp sau khi duyệt qua phần tử 3 của chuỗi thứ nhất: 1 2 0

Mảng dp sau khi duyệt qua phần tử 4 của chuỗi thứ nhất: 1 2 3

Độ dài lớn nhất của xâu con chung dài nhất: 3