

Solution LEXLCS

Đối với yêu cầu đầu tiên :

Gọi n là độ dài chuỗi X , m là độ dài chuỗi Y . Ta sẽ gọi $f[i][j]$ = một cái *pair*, *first* sẽ lưu độ dài chuỗi con chung dài nhất từ cuối đến i của chuỗi X và từ cuối đến j của chuỗi Y (nghĩa là ta qđđ từ cuối về), còn *second* sẽ lưu lại ký tự của gần nhất mà bạn lấy cho dãy con đó.

Công thức sẽ như sau :

Nếu $x[i] = y[j]$

$$f[i][j] = \{f[i + 1][j + 1].first + 1, x[i]\}$$

Ngược lại

if $(f[i + 1][j] < f[i][j + 1])$ $f[i][j] = f[i][j + 1]$;

else $f[i][j] = f[i + 1][j]$;

(Ta ưu tiên *first* trước rồi mới ưu tiên *second* trong so sánh $f[i + 1][j] < f[i][j + 1]$)

Còn về truy vết thì đơn giản, nó sẽ kiểu như thế này :

// res là kết quả

```
for (int i = 0 ; i < res ; i++)
{
    char l = f[a][b].second;
    kq = kq + l;
    while(x[a] != l) a++;
    a++;
    while(y[b] != l) b++;
    b++;
}
```

Đối với yêu cầu thứ 2 :

Gọi n là độ dài chuỗi X , m là độ dài chuỗi Y . Ta sẽ gọi $f[i][j]$ = một cái *pair*, *first* sẽ lưu độ dài chuỗi con chung dài nhất từ cuối đến i của chuỗi X và từ cuối đến j của chuỗi Y (nghĩa là ta qđđ từ cuối về), còn *second* sẽ lưu lại ký tự của gần nhất mà bạn lấy cho dãy con đó.

Công thức sẽ như sau :

Nếu $x[i] = y[j]$

$$f[i][j] = \{f[i+1][j+1].first + 1, x[i]\}$$

Ngược lại

```
{
    f1 = {f[i+1][j].first + 1, x[i]};
    f2 = {f[i][j+1].first + 1, y[j]};
    if(f1 < f2) f[i][j] = f1;
    else f[i][j] = f2;
}
```

(so sánh theo *first* trước rồi mới so sánh *second*)

truy vết sẽ như sau:

```
kq = "" ;
a = 1 ; b = 1;
while (a < m + 1 && b < n + 1)
{
    char l = f[a][b].c;
    kq = kq + l ;
    if (x[a] == y[b])
    {
        ++b;
        ++a;
        continue;
    }
    if (x[a] == l) a++;
    else b++;
}
for (; a <= m; a++) kq = kq + x[a];
for (; b <= n; b++) kq = kq + y[b];
```