- Quy hoạch động: Quy hoạch động là kĩ thuật được dùng khi có một công thức hoặc một (một vài) trạng thái bắt đầu. Một bài toán được tính bởi các bài toán nhỏ hơn đã được tìm ra từ trước, và kết quả các bài toán sẽ được lưu lại để những lần tính toán tiếp theo nếu cần đến những kết quả đó thì không cần tốn thêm thời gian thực hiện lại những bài toán này nữa. Nói cách khác, quy hoạch động bắt đầu từ việc giải các bài toán nhỏ, để từ đó từng bước giải quyết bài toán lớn hơn , và cuối cùng là bài toán lớn nhất ( bài toán ban đầu).

Khác với chia để trị, quy hoạch động thay vì gọi hàm đệ quy sẽ tính trước lời giải của các bài toán con và lưu vào bộ nhớ (thường là một mảng) và sau đó lấy bài toán con ở trong mảng để giải bài toán lớn.

Chia để trị: tiếp cận từ trên xuống

Quy hoạch động: tiếp cận từ dưới lên.

Gồm 3 bước:

* **Phân rã: chia bài toán cần giải thành những bài toán con nhỏ hơn đến mức có thể giải trực tiếp**
* **Giải các bài toán con và ghi nhận lời giải: lưu trữ lời giải của các bài toán con vào một mảng để sử dụng về sau**
* **Tổng hợp lời giải:**

**+ Tổng hợp lời giải của các bài toán con kích thước nhỏ hơn thành lời giải bài toán lớn hơn**

**+ Tiếp tục cho đến khi thu được lời giải của bài toán xuất phát**

1. **Nêu bài toán.**

Cho hai xâu S độ dài m và xâu T độ dài n, tìm xâu có độ dài lớn nhất là xâu con của cả hai xâu S và T.

Công thức tìm xâu con chung dài nhất của 2 xâu A, B:

* Nếu A[i]=B[j], F[i][j]=F[i−1][j−1]+1
* Ngược lại, F[i][j]=max{F[i−1][j],F[i][j−1]}

1. **Mô tả thuật toán**

Thuật toán trực tiếp để giải bài toán đặt ra là: Duyệt tất cả các dãy con của dãy X và kiểm tra xem mỗi dãy như vậy có là dãy con của dãy Y, và giữ lại dãy con dài nhất. Mỗi dãy con của X tương ứng với dãy chỉ số <i1,i2, …, ik> là tập con k phần tử của tập chỉ số {1, 2, …, m}, vì thế có tất cả 2m dãy con của X.Như vậy thuật toán trực tiếp đòi hỏi thời gian hàm mũ và không thể ứng dụng được trên thực tế. Ta xét áp dụng quy hoạch động để xây dựng thuật toán giải bài toán này.

Phân rã . Với mỗi 0 < i < m và 0 < j < n xét bài toán C(i,j); tính C[i,j] là độ dài của dãy con chung dài nhất của hai dãy.

Xi=<x1,x2,…,xi> và yi=<y1,y2,…yi>

Như vậy ta đã phân bài toán cần giải ra thành (m+1)x(n+1) bài toán con. Bản thân bài toán xuất phát là bài toán con có kích thước lớn nhất C(m,n).

Tổng hợp lời giải.

Lấy i ký tự đầu tiên từ xâu thứ nhất và j ký tự đầu tiên từ xâu thứ hai và tìm độ dài xâu chung dài nhất giữa 2 xâu con được lấy ra đó. Dễ dàng thấy được rằng, lời giải của mỗi bài toán con sẽ phụ thuộc vào i và j, arr(i, j). Và bài toán lớn sẽ được giải bằng cách lần lượt giải các bài toán con lần lượt từ arr(0, 0) và tăng dần độ dài xâu được lấy ra cho đến khi chúng ta lấy ra toàn bộ xâu của đề bài.

Chúng ta hãy bắt đầu lần lượt các bài toán con. Đương nhiên, nếu một trong hai xâu là rỗng thì xâu con chung của chúng cũng rỗng. Vậy arr(0, j) = arr(i, 0) = 0. Nếu cả i và j đều dương, chúng ta cần suy xét một vài trường hợp.

1. Nếu ký tự cuối cùng của xâu thứ nhất không có mặt trong xâu con chung dài nhất, nó có thể bị bỏ qua mà không ảnh hưởng gì đến kết quả. Công thức ở đây sẽ là arr(i, j) = arr(i - 1, j).
2. Tương tự như trường hợp trên, ký tự cuối cùng của xâu thứ hai không ảnh hưởng đến kết quả thì arr(i, j) = arr(i, j - 1).
3. Trường hợp cuối cùng, nếu hai ký tự cuối cùng của hai xâu x1, x2 đều có mặt trong xâu con chung dài nhất. Dĩ nhiên là hai ký tự này phải là một thì điều này mới xảy ra, tức là x1 == x2. Trong trường hợp này, khi xoá đi bất cứ một ký tự nào trong hai ký tự đó đều khiến xâu con chung dài nhất ngắn đi 1 ký tự. Vậy rõ ràng là arr(i, j) = arr(i - 1, j - 1) + 1.

Thủ tục

Procedure LCS(X,Y);

begin

for i :=1 to m do arr[i,0]:=0;

for j: =1 to n do arr[0,j:=0;

for i: =1 to m do

for j: = 1to n do

if xi = yi then

begin

arr[i,j]:=arr[i-1,j-1]+1;

end

else

if arr[i-1,j] ≥ arr[i,j-1] then

begin

arr[i,j]:=arr[i-1,j];

end

else

begin

arr[i,j]:=arr[i,j-1];

end;

end;

- Để truy vết lại chuỗi con dài nhất ta sử dụng thuật toán back tracking với thủ tục như sau:

Procedure Back\_tracking:

i=n;

j=m;

While (arr[i][j]!=0) do

If Xi-1==Yi-1 :

retString += Xi-1;

i--;j--;

else:

If Xi-1>=Yi-1 : i--;

Else : j--;

Reverse(retString);

end

1. **Kiểm thử**
2. Bộ thứ nhất

|  |  |
| --- | --- |
| CHUOICHUNG  HUONGDANBAI | HUONG  maxLen = 5 |

X = CHUOICHUNG

Y = HUONGDANBAI

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| j |  |  | H | U | O | N | G | D | A | N | B | A | I |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | H | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | U | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 4 | O | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | I | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 |
| 6 | C | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 |
| 7 | H | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 |
| 8 | U | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 |
| 9 | N | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 10 | G | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

1. Bộ thứ hai

|  |  |
| --- | --- |
| KHOAHOC  HOAHONG | HOAHO  maxLen = 5 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| j |  |  | H | U | A | H | O | N | G |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | K | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | H | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | O | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 4 | A | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 |
| 5 | H | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 4 |
| 6 | O | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 7 | C | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |

Kết quả : HOAHO

1. Bộ thứ ba

|  |  |
| --- | --- |
| MONLAPTRINH  CHUONGTRINH | ONTRINH  maxLen = 7 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| j |  |  | C | H | U | O | N | G | T | R | I | N | H |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | M | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | O | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | N | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 4 | L | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 5 | A | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 6 | P | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | T | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 |
| 8 | R | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| 9 | I | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 4 | 5 | 5 | 5 |
| 10 | N | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 4 | 5 | 6 | 6 |
| 11 | H | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |

Kết quả: ONTRINH

1. Bộ thứ tư

|  |  |
| --- | --- |
| BAOCAODAIQUA  CHUALAMXONG | CUA  maxLen = 3 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| j |  |  | C | H | U | A | L | A | M | X | O | N | G |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | B | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | A | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | O | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| 4 | C | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 5 | A | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 |
| 6 | O | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 |
| 7 | D | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 8 | A | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 9 | I | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 10 | Q | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 11 | U | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 12 | A | 0 | 1 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |

Kết quả: CUA

1. Bộ thứ năm

|  |  |
| --- | --- |
| SANGTAOMOI  TAOMOTBIEN | TAOMOI  maxLen = 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| j |  |  | T | A | O | M | O | T | B | I | E | N |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | A | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | N | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 4 | G | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 5 | T | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 6 | A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | O | 0 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 8 | M | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 9 | O | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| 10 | I | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 5 | 6 | 6 | 6 |

Kết quả: TAOMOI

1. **Code**

|  |
| --- |
| #include <bits/stdc++.h>  #include <string>  using namespace std;  const int N = 100;  string str1 = "";  string str2 = "";  int arr[N][N];  void LCS(int n, int m)  {      for (int i = 0; i <= n; i++)          arr[i][0] = 0;      for (int j = 0; j <= m; j++)          arr[0][j] = 0;      for (int i = 1; i <= n; i++)          for (int j = 1; j <= m; j++)              if (str1[i - 1] == str2[j - 1])                  arr[i][j] = arr[i - 1][j - 1] + 1;              else              {                  if (arr[i - 1][j] >= arr[i][j - 1])                      arr[i][j] = arr[i - 1][j];                  else                      arr[i][j] = arr[i][j - 1];              }  }  string Back\_tracking(int n, int m)  {      int i = n;      int j = m;      string ret = "";      while (i > 0 && j > 0) {          if (str1[i - 1] == str2[j - 1]) {              ret = str1[i - 1] + ret;              i--;              j--;          } else if (arr[i - 1][j] > arr[i][j - 1]) {              i--;          } else {              j--;          }      }      return ret;  }  int main()  {      freopen("input.txt", "r+", stdin);      freopen("output.txt", "w+", stdout);      getline(cin, str1);      getline(cin, str2);      time\_t st,ed;      st = clock();      int n = str1.length();      int m = str2.length();      LCS(n, m);      string ret = Back\_tracking(n,m);      ed = clock();        cout << str1 << endl           << str2 << endl;      for (int i = 0; i <= n; i++)      {          for (int j = 0; j <= m; j++)          {              cout << arr[i][j] << " ";          }          cout << endl;      }      cout<<"Max LCS length: "<<arr[n][m]<<endl;      cout<<"LCS: "<<ret<<endl;      cout<<"Time: "<<setprecision(8)<<(double) ed-st<<endl;  } |

1. **Đánh giá thuật toán**

Thuật toán dùng quy hoạch động có độ phức tạp là O(mn).