

Bài toán chuỗi con đối xứng dài nhất

Một chuỗi được gọi là đối xứng (palindrome) nếu như khi đọc chuỗi này từ phải sang trái cũng thu được chuỗi ban đầu.

Yêu cầu: tìm một chuỗi con đối xứng dài nhất của một chuỗi s cho trước. ***Chuỗi con là chuỗi thu được khi xóa đi một số ký tự từ chuỗi ban đầu.***

Dữ liệu vào

Gồm một dòng duy nhất chứa chuỗi S , chỉ gồm những chữ cái in thường.

Kết quả

Gồm một dòng duy nhất là một chuỗi con đối xứng dài nhất của chuỗi S . Nếu có nhiều kết quả, chỉ cần in ra một kết quả bất kỳ.

Giới hạn

Chuỗi S có độ dài không vượt quá 2000.

Ví dụ

Dữ liệu vào

lmevxeyzl

Kết quả

level

Ta sẽ chuyển bài toán về một bài toán quy hoạch động cơ bản là: [Bài toán tìm chuỗi con chung dài nhất](#)

Với dữ liệu vào là $S1$.

Ta tạo chuỗi $S2$ là chuỗi ngược của $S1$ bằng cách chép các phần tử của chuỗi $S1$ vào chuỗi $S2$ theo thứ tự ngược lại.

Sau đó ta sẽ tìm chuỗi con chung dài nhất của $S1$ và $S2$.

Ta chỉ cần tìm chuỗi con chung dài nhất của một phần của $S1$ và nghịch đảo phần còn lại, tức là ta chỉ xét một phần của bảng phương án với $i+j \leq \text{chiều dài của } S1$.

Ví dụ:

$S1 = \text{lmevxeyzl}$

Ta có bảng phương án

S1 \ S2	0	1	2	3	4	5	6	7	8	9
			m	e	v	x	e	y	z	
0		0	0	0	0	0	0	0	0	x
1		0	1	1	1	1	1	1	x	x
2	z	0	1	1	1	1	1	x	x	x
3	y	0	1	1	1	1	x	x	x	x
4	e	0	1	1	2	2	x	x	x	x
5	x	0	1	1	2	x	x	x	x	x
6	v	0	1	1	x	x	x	x	x	x
7	e	0	1	x	x	x	x	x	x	x
8	m	0	x	x	x	x	x	x	x	x
9		x	x	x	x	x	x	x	x	x

Khi đó chuỗi con chung dài nhất là *le* của $S1_4 = \text{"lmev"}$ và $S2_4 = \text{"lzye"}$ (hoặc $S1_3 = \text{"lme"}$ và $S2_5 = \text{"lzyex"}$)

Khi ta truy vết để tìm chuỗi con chung ta sẽ kiểm tra xem chuỗi đối xứng của chúng ta là lẻ hay chẵn (số kí tự).

- Nếu $i+j = \text{chiều dài của } S1$, tức là 2 kí tự đối xứng đứng liên tiếp nhau trong $S1$, vì vậy chuỗi đối xứng là chẵn.
- Ngược lại, tức là mọi $i+j < \text{chiều dài của } S1$, thì trong $S1$, có các kí tự xen giữa hai kí tự đối xứng, nên chuỗi đối xứng là lẻ. Trong ví dụ trên thì có hai kí tự *v* và *x* xen giữa chuỗi đối xứng

Với chuỗi đối xứng chẵn ta chỉ việc sao chép lại nửa sau dựa vào nửa đầu.

Còn chuỗi lẻ ta sẽ chọn thêm một kí tự xen giữa. Theo ví dụ trên, có thể chọn *v* hoặc *x*. Như vậy chuỗi đối xứng dài nhất sẽ là *level* hoặc *lexel*

Cài đặt bài toán bằng ngôn ngữ C++

```
#include <iostream>
#include <fstream>
#include <string.h>
using namespace std;
#define Input "doixung.inp"
```

```

#define Output "doixung.out"

char S1[2000],S2[2000],S[2000];
int B[2001][2001],T[2001][2001];
int L,MaxP;

void GetInput()
{
    ifstream fi(Input);
    fi>>S1;
    fi.close();
    L=strlen(S1);
    for (int i=0; i<L;i++) S2[L-i-1]=S1[i];
}

void PutOutput()
{
    ofstream fo(Output);
    fo<<S;
    fo.close();
}

void Optimize()
{
    int i,j;
    for (i=0; i<L; i++)
        B[0][i]=B[i][0]=0;
    for (i=1; i<=L; i++)
        for (j=1; j<=L; j++)
            if (i+j<=L)
            {
                if (S1[i-1]==S2[j-1])
                {
                    B[i][j]=B[i-1][j-1]+1;
                    T[i][j]=0;
                }
                else
                    if (B[i-1][j]>B[i][j-1])
                    {
                        B[i][j]=B[i-1][j];

```

```

        T[i][j]=1;
    }
    else
    {
        B[i][j]=B[i][j-1];
        T[i][j]=-1;
    }
}
}

```

```

void Trace()
{
    int maxP=0;
    int i,j;
    for (i=1; i<=L; i++)
        if (B[i][L-i]>maxP)
        {
            maxP=B[i][L-i];
            j=L-i;
        }
    i=L-j;
    int k=i;
    int Ls=maxP;
    bool even=false;
    while (i>0 && j>0)
    {
        if (T[i][j]==0)
        {
            if (i+j==L) even=true;
            S[--Ls]=S1[i-1];
            i--;
            j--;
        }
        else if (T[i][j]==1)
            i--;
        else
            j--;
    }
}

```

```

j=maxP-1;

```

```
if (!even)
    S[maxP++] = S1[k];

while (j >= 0)
    S[maxP++] = S[j--];
S[maxP] = 0;

}

int main()
{
    GetInput();
    Optimize();
    Trace();
    PutOutput();
    return 0;
}
```