

ЗИМНИ МАТЕМАТИЧЕСКИ ПРАЗНИЦИ – РУСЕ’ 2006

Вход: На първия ред на стандартния вход е зададен броят P на буквите, за които $V(a) \neq 100$. На всеки от следващите P реда е зададена по една двойка: $a \ V(a)$, разделени с интервал. Следва описанието на $D(a, b)$. На отделен ред е зададен броят Q на двойките знаци, чиято различност не е 100. На всеки от следващите Q ($Q \leq 100$) реда има по една тройка $a \ b \ D(a, b)$, разделени с по един интервал. Следват два реда с низовете α и β . Последният ред на стандартния вход е празен, т.е. след β има знак за нов ред. Дължините на α и β са между 1 и 1023 знака.

Изход: Разстоянието между α и β да се изведе на стандартния изход.

ПРИМЕР:

Вход:	Изход:
1 _ 50 3 O 0 50 1 I 50 A 4 10 BIOGRAPHY B10_GR4PHIE	3 6 0

Решение:

Това е модификация на задачата за намиране на минимално разстояние при редактиране на низове (minimum edit distance). Решението и тук става с динамично оптимиране.

Оригиналната задача (без тежести на операциите insert, update, delete) се решава с помощта на двумерна $(m+1) \times (n+1)$ табличка t , в която t_{ij} е разстоянието между префиксите $a_0 a_1 \dots a_{i-1}$ и $b_0 b_1 \dots b_{j-1}$:

$$\begin{cases} t_{00} = 0 \\ t_{i0} = t_{i-1,0} + 1 = i \\ t_{0j} = t_{0,j-1} + 1 = j \\ t_{ij} = \min\{t_{i-1,j} + 1, \quad t_{i,j-1} + 1, \quad t_{i-1,j-1} + \varepsilon(a_{i-1}, b_{j-1})\} \end{cases}$$

където

$$\delta(x, y) = \begin{cases} 0, & x \neq y \\ 1, & x = y \end{cases}$$

Решението на цялата задача е t_{mn}

t	j=0	j=1	j=2		j=n
i=0					
i=1					
i=2					
				t_{ij}	
i=m					решение

При варианта с тежести рекурентната връзка изглежда така:

ЗИМНИ МАТЕМАТИЧЕСКИ ПРАЗНИЦИ – РУСЕ’ 2006

$$\begin{cases} t_{00} = 0 \\ t_{i0} = t_{i-1,0} + V(a_{i-1}) \\ t_{0j} = t_{0,j-1} + V(b_{j-1}) \\ t_{ij} = \min\{t_{i-1,j} + V(a_{i-1}), \quad t_{i,j-1} + V(b_{j-1}), \quad t_{i-1,j-1} + D(a_{i-1}, b_{j-1})\} \end{cases}$$

```
#include <stdio.h>
#include <string.h>
```

```
#define MAX_N 1024
```

```
char v[256];    // Values are between 0 and 100
char d[256][256]; // Values are between 0 and 100
char alpha[MAX_N]; // No need for "unsigned char" - acceptable chars have ascii codes less than 128
char beta[MAX_N];
int t[MAX_N][MAX_N];
```

```
int min3(int x, int y, int z) {
    return (x < y)
        ? ((x < z) ? x : z)
        : ((y < z) ? y : z);
}
```

```
void main() {
```

```
    // Input: visibility
    memset(v, 100, sizeof(v));
    int p;
    scanf("%d", &p);
    for (int i = 0; i < p; i++) {
        unsigned char s[1];
        int x;
        scanf("%s%d", s, &x);
        int ch = s[0];
        v[ch] = x;
    }
```

```
    // Input: difference
    memset(d, 100, sizeof(d));
    for (int i = 0; i < (sizeof(d) / sizeof(d[0])); i++) {
        d[i][i] = 0;
    }
    int q;
    scanf("%d", &q);
    for (int i = 0; i < q; i++) {
        unsigned char s1[1];
        unsigned char s2[1];
        int x;
        scanf("%s%s%d", s1, s2, &x);
        int ch1 = s1[0];
```

ЗИМНИ МАТЕМАТИЧЕСКИ ПРАЗНИЦИ – РУСЕ’ 2006

```
int ch2 = s2[0];
d[ch1][ch2] = d[ch2][ch1] = x;
}

// Input: alpha and beta
scanf("%s%s", alpha, beta);
int nAlpha = strlen(alpha);
int nBeta = strlen(beta);

// Calculate distance
t[0][0] = 0;
for (int i = 1; i <= nAlpha; i++) {
    t[i][0] = t[i - 1][0] + v[alpha[i - 1]];
}
for (int j = 1; j <= nBeta; j++) {
    t[0][j] = t[0][j - 1] + v[beta[j - 1]];
}
for (int i = 1; i <= nAlpha; i++) {
    for (int j = 1; j <= nBeta; j++) {
        int a = alpha[i - 1];
        int b = beta[j - 1];
        t[i][j] = min3(
            t[i - 1][j - 1] + d[a][b], // update
            t[i][j - 1] + v[b],        // insert
            t[i - 1][j] + v[a]          // delete
        );
    }
}

// Output
printf("%d\n", t[nAlpha][nBeta]);
}
```

Задача В2. Y1984

Годината е 0x1984 (или 6532 по старото десетично летоброене). На света са останали само три, постоянно воюващи помежду си държави, и, като служител на тайните служби на Евразия, ти имаш задачата да шпионираш тайните агенти на останалите две. Знае се, че на територията на страната има **N** чужди тайни агента, с кодови имена Агент 0 ... Агент **N-1**, но не се знае кой за коя от вражеските държави работи. За щастие разполагаш със записи на телефонни разговори между някои от тях. От всеки записан разговор става ясно дали говорещите са приятели (от една и съща държава) или врагове. От вас се иска да напишете програма **Y1984**, с помощта на която да можете да отговаряте на въпроса: „Приятели или врагове са агентите **X** и **Y**”. Възможните отговори са „приятели”, „врагове” или „недостатъчно информация”.

Вход: На първите два реда от стандартния вход са дадени числата **N** и **M** – броят на тайните агенти и броят на записаните телефонни разговори, съответно. На всеки от следващите **M** реда има по една тройка числа **A**, **B**, **R**, разделени с по един интервал, където **A** и **B** са номера на агенти, а **R** е връзката между тях. **R**=0 за „приятели”, и **R**=1 за „врагове”. Следват въпросите. На отделен ред е даден броят им **Q**. На всеки от следващите **Q** реда има