

Full Solution: $N \leq 20$

Do dynamic programming on subsets. The DP state for a subset $S = \{c_1, c_2, \dots, c_n\}$ of the cowphabet is $\min_p (\text{evaluate}(p))$ over all $n!$ permutations p of S . The DP transition for all nonempty S is as follows:

$$\text{dp}[S] = \min_{c_j \in S} \left(\text{dp}[S \setminus \{c_j\}] + \sum_{c_k \in S} (\# \text{ occurrences of } c_j c_k) \right)$$

\setminus	补集		
	减; 除去	$A \setminus B$ 表示所有属于 A 但不属于 B 的元素的集合。	$\{1, 2, 3, 4\} \setminus \{3, 4, 5, 6\} = \{1, 2\}$
	集合 论	(有的地方记作 $\complement_A B$)	$\complement_U A = \{x x \in U \text{ and } x \notin A\}$

0	1	2	3	4	5	6	7	8	9
m	i	l	d	r	e	d	r	e	e

occ[m->m] -> occ[m][m] occ[20][20]

occ[m->m]=0 occ[d->m]=0
occ[m->i]=1 occ[d->r]=2
occ[m->l]=0 occ[d->e]=0
... ...

0	1	2	3	4	5	6	7	8
m	i	l	d	r	e			

dp[0]=1表示不做任何组合的时候答案是1(最小次数是1)

dp[m] =dp[000001]
dp[mi] =dp[000011]
...
dp[me] =dp[100001]
...
dp[dre] =dp[111000]
...
dp[mil] =dp[000111]
...
dp[mildre]=dp[111111]

2⁵

dp[m]=dp[0]+occ[m->m] ans

dp[i]=dp[0]+occ[i->i]

dp[mi]=min { dp[i]+(occ[m->m]+occ[m->i])
 dp[m]+(occ[i->m]+occ[i->i])
 ...

缺少m的集合

m的所有情况

ans=dp[mildre] =min { dp[ildre]+(occ[m->m]+occ[m->i]+occ[m->l]+occ[m->d]+occ[m->r]+occ[m->e])
dp[mldre]+(occ[i->m]+occ[i->i]+occ[i->l]+occ[i->d]+occ[i->r]+occ[i->e])
dp[milre]+(occ[d->m]+occ[d->i]+occ[d->l]+occ[d->d]+occ[d->r]+occ[d->e])
dp[milde]+(occ[r->m]+occ[r->i]+occ[r->l]+occ[r->d]+occ[r->r]+occ[r->e])
dp[mildr]+(occ[e->m]+occ[e->i]+occ[e->l]+occ[e->d]+occ[e->r]+occ[e->e])

缺少e的集合

e的所有情况

```
bitset<8> b0("0011010");
bitset<8> b2("0000010");
bitset<8> b3("0001000");
bitset<8> b4("0100000");
cerr << (b0 & b2) << ", 是否包含b2\n"; ..... //是否包含b2
cerr << (b0 & b3) << ", 是否包含b3\n"; ..... //是否包含b3
cerr << (b0 & b4) << ", 是否包含b4\n"; ..... //是否包含b4
cerr << (b0 ^ b2) << ", 排除b2\n"; ..... //排除b2
```

```
2021/02/02 22:27:24 Launching "a.exe"
00000010, 是否包含b2
00001000, 是否包含b3
00000000, 是否包含b4
00011000, 排除b2
Execution time: 0 ms
```