

知识精炼（六）



主讲人：邓哲也



Codeforces 827C

给定一个只包含 A, T, C, G 的字符串，有如下两种操作：

(1) 修改某个位置的字母

(2) 给定一个字符串 s ($|s| \leq 10$)，生成一个由 s 重复组成的新串： $sssss\dots$ 。询问 $[L, R]$ 中有几个字母跟这个新的字符串对应。

$n, q \leq 100000$

input

ATGCATGC

4

2 1 8 ATGC

2 2 6 TTT

1 4 T

2 2 6 TA

output

8

2

4

Codeforces 827C

样例字符串 $T = \text{ATGCATGC}$

注意到每次询问的字符串 s 的长度不超过 10, 记 $L = |s|$

假设每次询问的区间都是 $[1, n]$ 。

我们只对 T 中的前 L 个字符考虑。

对于 $s[i]$, 我们要判断的是 $T[i]$, $T[i+L]$, $T[i+2L]$, \dots , $T[i+kL]$ ($i + kL \leq n$) 是否等于 $s[i]$ 即可。

如果想查询区间 $[1, q]$ 的匹配情况, 只要判断 $T[i]$, $T[i+L]$, \dots , $T[i+kL]$ ($i + kL \leq q$) 即可。

Codeforces 827C

对于每种可能的 L （一共 10 种情况）：

对于每个 i ($1 \leq i \leq L$)：

对于每个可能的 $s[i]$ ($s[i] \in \{A, C, G, T\}$)：

我们都建立一个长度为 n 的树状数组，第 j 位表示 $T[j]$ 和 $s[i]$ 是否相等。

其中 j 只取 $\{i, i+L, i+2L, \dots, i+kL\}$

总共预处理 $4 * (1 + 2 + 3 + \dots + 10) = 220$ 个长度为 n 的树状数组。

现在对于一般的询问 $[1, r]$ ，长度为 L 的 s ：

只要对 $T[1..r]$ 的前 L 个字符考虑即可。

对于 $T[1]$ ，我们先找到 i

$$i = \min \{ 1 - kL \mid 1 - kL > 0 \}$$

然后在 $(L, i, s[1])$ 对应的树状数组上查询 $(1, r)$ 上有几个 1 即可。

Codeforces 827C

```
#define N 100010

int bit[4][11][11][N], has[128], n;
char s[N];

void add(int p, int c, int v) {
    for (int x = p; x <= n; x += x & -x)
        for (int j = 1; j <= 10; j++)
            bit[c][j][p % j][x] += v;
}
```

Codeforces 827C

```
int ask(int c, int len, int mo, int x){
    int ans = 0;
    for (;x;x -= x & -x) ans += bit[c][len][mo][x];
    return ans;
}

int query(int c, int len, int mo, int l, int r){
    return ask(c, len, mo, r) - ask(c, len, mo, l - 1);
}
```

Codeforces 827C

```
int main() {
    has['A'] = 0; has['T'] = 1; has['C'] = 2; has['G'] = 3;
    scanf("%s", &s[1]);
    n = strlen(s+1);
    for (int i = 1; i <= n; ++i) add(i, has[s[i]], 1);
    char ch[20];
    int q, a, b, c;
    scanf("%d", &q);
    while (q--) {
        scanf("%d", &a);
        if (a == 1) {
            scanf("%d %s", &b, ch);
            add(b, has[s[b]], -1);
            add(b, has[ch[0]], 1);
            s[b] = ch[0];
        }
    }
}
```


Codeforces 827C

```
    } else {  
        scanf("%d %d %s", &b, &c, ch);  
        int len = strlen(ch), ans = 0;  
        for (int i = 0; i < len; i ++)  
            ans += query(has[ch[i]], len, (i + b) % len, b, c);  
        printf("%d\n", ans);  
    }  
}  
return 0;  
}
```

下节课再见