3 QOJ 850 Edit Distance Yet Again

3.1 题目大意

字符串 S,T 的编辑距离为 S 通过以下三种操作变为 T 的最少步数：在任意位置插入任意字符；删除任意位置的字符；将任意位置的字符替换为任意字符。

给定长为 n,m 的两字符串 S,T 和整数 k，问 S 和 T 的编辑距离是否不超过 k。若不超过，输出一种最少步数的方案。

3.2 数据范围

本题单个测试点内有 t 组数据。

保证 $t \le 100$，$0 \le k \le 1000$，$n,m \le 10^6$，$\sum(n+m) \le 10^7$。

3.3 解题过程

3.3.1 算法 1

暴力 DP。设 $f_{i,j}$ 表示 $S[1:i]$ 与 $T[1:j]$ 的编辑距离。

有转移：$f_{i,j} = \min\{f_{i-1,j-1} + [S_i \neq T_j], f_{i-1,j} + 1, f_{i,j-1} + 1\}$。

时间复杂度 $\mathcal{O}(\sum nm)$，难以通过。

3.3.2 算法 2

考虑利用 k 较小的性质。注意到在上面的 DP 中，$f_{i,j} \ge |i-j|$，因此我们只需要保留 $|i-j| \le k$ 的状态进行 DP 即可。

设 $f_{i,j}$ 表示 $S[1:i]$ 与 $T[1:(i+j)]$ 的编辑距离，其中 $|j| \le k$。

转移：$f_{i,j} = \min\{f_{i-1,j} + [S_i \neq T_{i+j}], f_{i-1,j+1} + 1, f_{i,j-1} + 1\}$。

时间复杂度优化至 $\mathcal{O}(k\sum\min(n,m))$，依然难以通过。

3.3.3 算法 3

注意到，$f_{i,j} > k$ 的状态依然是没用的，而且 $f_{i,j} \ge f_{i-1,j}$。这启发我们交换状态的下标与值：设 $g_{i,j}$ 为最大的 x 满足 $f_{x,j} \le i$，其中 $i \le k$。

那么根据上面的 DP 过程，可以写出从 $g_{i,j}$ 转出的过程：

1. 在 $S_{g_{i,j}+1} = T_{g_{i,j}+1+j}$ 的情况下，重复 $g_{i,j} \leftarrow g_{i,j} + 1$。

2. $g_{i,j}+1$ 转移到 $g_{i+1,j}$ 和 $g_{i+1,j-1}$。$g_{i,j}$ 转移到 $g_{i+1,j+1}$。

这里和前面的直接 DP 算法一样，在 DP 的过程中同时记录转移点，最后仍然可以还原方案。

时间复杂度依旧是 $\mathcal{O}(k\sum\min(n,m))$，瓶颈在 $g_{i,j}$ 自增的过程，其余部分是 $\mathcal{O}(k^2)$ 的。

3.3.4 算法 4

考虑优化 $g_{i,j}$ 自增的过程。这本质上其实是 $\mathcal{O}(k^2)$ 次询问 S 某后缀和 T 某后缀的 LCP。

考虑二分，这样问题变为需要快速判断 S 的某个子段和 T 的某个子段是否相同，可以通过哈希和预处理做到 $\mathcal{O}(n+m)-\mathcal{O}(1)$ 判断。

总时间复杂度 $\mathcal{O}(\sum(n+m)+tk^2\log n)$。由于本题时间限制开的非常宽松，该做法是可以通过的。

3.3.5 算法 5

可以对串 $R=S+T$ 构建后缀数组，在求出其 height 数组后，该询问变为 RMQ 问题。用 ST 表可以做到 $\mathcal{O}(\sum(n+m)\log n+tk^2)$。

更进一步的，使用线性构建后缀数组的算法，再配合 $\mathcal{O}(n)-\mathcal{O}(1)$ RMQ，本题可以做到时间复杂度 $\mathcal{O}(\sum(n+m)+tk^2)$。

3.4 参考资料

Edit distance - Wikipedia