Lab2

姓名：徐兆恺
学号：23300240008

Task1

解决思路

因为序列极长，推断每个匹配好的序列都很长（大于30）.对于每个匹配的序列，若是他的长度大于等于30，一定可以被分成若干段30长度+一段30-59长度的匹配

所以只需要找出来所有长度在30-60的hash完全匹配串，就可以"拼凑"出所有较长的匹配串

如何找到总和最长串?

考虑动态规划，dp[i] = max(dp[j] + (i-j)) if(query[j:i]在ref中有完全匹配)

最后进行回溯，可以找到所有匹配的位置，输出答案

伪代码

```
for i from 1 to m
    if i < Llen //长度不够30，不可能有匹配
        continue
    for j from i-Llen to i-Rlen
        if find(j+1, i) == true // 在ref中找到了匹配
            dp[i] = max(dp[i], dp[j] + (i-j))
```

时空复杂度分析

len在长度30-60之间，所以内部每次只会循环30次，find可以用HASH+HASH表进行 O(1) 的判断查找，所以总共为 O(30*n) 的时间复杂度，接近线性

Task2

解决思路

序列长度较短，并且在进行了task1相同算法尝试后表现不佳，推断为每隔一小段就会有随机的小突变产生，这样总共就没有大的突变但是无法用30左右的串进行拼凑
考虑暴力贪心， DP 与task1相同，只不过find函数从完全hash匹配到循环暴力判断是否有 0.9 以上的匹配度，如果有的话找到分最高的一个进行匹配，分数加进 DP
最终在多次实验后，发现长度为恰好100时表现最佳

伪代码

```
for i from 1 to m
    if i < Llen //长度不够30，不可能有匹配
        continue
    for j from i-Llen to i-Rlen
        G = find(j+1, i)
        if G.flag == true // 在ref中找到了匹配
            dp[i] = max(dp[i], dp[j] + G.score)
```

时空复杂度分析

发现多个长度混合时表现不佳，最终Llen = Rlen = 100 时表现最好，但是find的复杂度为 O(len*n) 最终会循环m次，最坏复杂度为 O(100nm) ,但是考虑在find中加入剪枝，若是不匹配率超过 0.1 就结束循环进行下次匹配，况且匹配的概率很小，平均复杂度接近 O(10nm)

运行结果

由于输出文档过大（1.out有24KB的大小），均上传至github

Task1 评分 29823
Task2 评分 2078

网址：https://github.com/tuihuademing2/DNA