碱基配对题解

对于前 10%的数据，随便写一写过了；

对于前 35%的数据，枚举匹配的位置 p，再枚举 B 中的一个字符，用 O(k)的时间判断是否存在相同的字符，时间复杂度 $O(n^3)$；

对于前 65%的数据，发现判断是否存在相同的字符可以用前缀和优化，时间复杂度 $O(n^2)$；

对于另 5%的数据，输出 n-m+1 即可；

对于 100%的数据，考虑枚举每个字符，用 a_i 表示 A_{i-k} 到 A_{i+k} 中是否出现过这个字符，用 b_i 表示 B_i 是否为这个字符，则在第 p 位能匹配到的数量为 $b_0*a_p+b_1*a_{p+1}+...+b_{m-1}*a_{p+m-1}$，不难发现反转 a 数组后即为一个卷积。于是可以用 FFT 或 NTT 快速求出这个字符在每一位能匹配到的数量，之后枚举每一位并判断 4 种字符能匹配到的数量和是否为 m 即可，时间复杂度 O(nlogn)。

《小凯的疑惑》题解

EricHuang

不知道大家看到小凯是否回忆起了 NOIP2017 呢？

序

显然n个点没有任何意义，只需要关心$[0,2^{\mathrm{C}}-1)$中那些数出现过即可，出现多次的数无论有多少操作都可以用0边连起来。其次我们只需要关心询问的前缀和（因为有后效性）而且因为是在取模意义下所以询问最多有2^{C}种。

子任务一

你会不会 Kruskal，Prim 或者至少一种最小生成树算法呢？会的话就能拿到9分了呢。复杂度$\mathrm{O}(C2^{3C})$。

子任务二

经典的异或生成树问题，可以把点按照权值分成最高为为0的和最高为1的两部分，显然两部分内的边（最高位为0）比跨越两部分的边（最高位为1）都要小。所以可以分治两边然后将左侧每个值加入一个 01Trie 然后右边每个点询问一下最小异或和就好了，注意特判一边没有点的情况。复杂度$\mathrm{O}(\mathrm{QC}^2 2^C)$。

子任务三

这询问的特性使得每次询问是将上面我们所说的分治树从叶子往上第五层的所有子树向后 shift 一下，可以看出由于最下面五层的所有子树是不会变的所以可以预处理出最下面五层每个子树答案以及两两之间合并（仅二进制最后五位）的结果，之后对于每个询问可以像子任务二一样合并上去。复杂度$\mathrm{O}(\mathrm{C}^2 2^{2C-10})$。

子任务四

上一个子任务给了我们一些提示，这个字任务中，由于$\mathrm{C}\leq 11$，所以我们可以将询问分为4类，即对2^2取模后同一种余数分一类，然后就像上面那样搞就好了，复杂度$\mathrm{O}(C^2 2^{2C-2})$。

子任务五

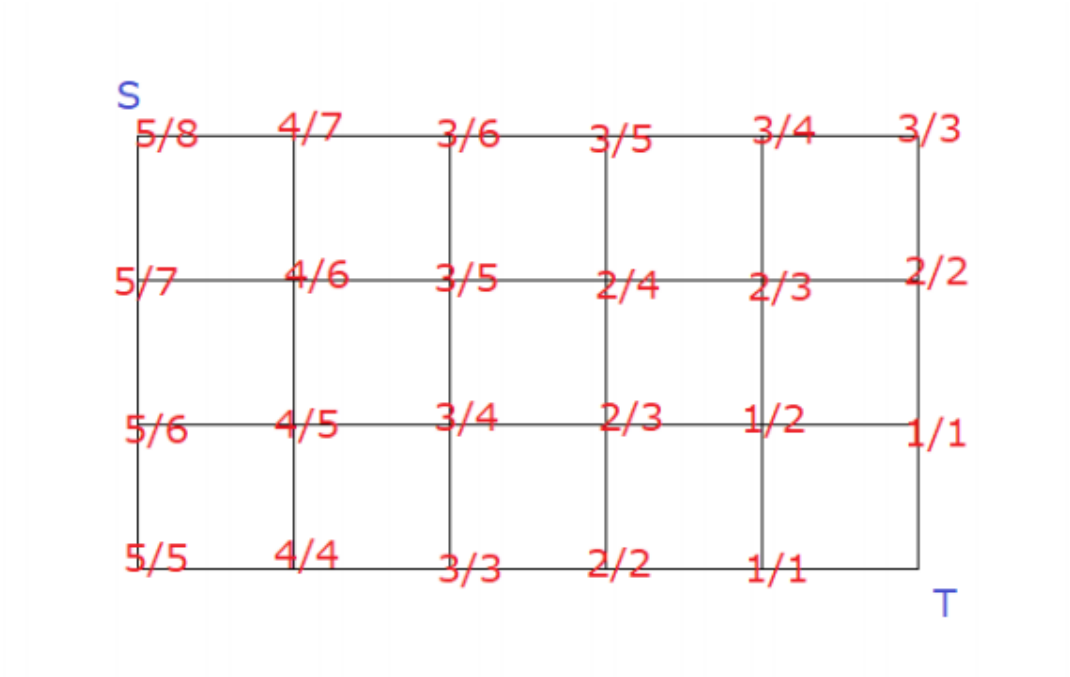
依然是仿照子任务三的方法，假设我们把询问按二进制位分割成$[0,\mathrm{B}),[\mathrm{B},\mathrm{C}]$两部分，如果我们能快速的算出对于所有的$2^{\mathrm{B}}$种情况下每个深度为B的子树的答案和两两之间合并最后B位的结果就可以套用子任务三了。如何计算呢？这里我们$[0,\mathrm{B}]$这一段二进制位再分两部分$[0,\mathrm{A}),[\mathrm{A},\mathrm{B}]$。对于一个深度为A的子树，其叶子最多有$2^{\mathrm{A}}$个，每个叶子可有可无，则一共有$2^{2^{\mathrm{A}}}$种子树，可以暴力处理出来，同时求出他们两两之间的答案，进而通过类似算法三的方法算出每个深度为B的子树的答案和两两之间合并最后B位的结果，进而继续用算法三求解。复杂度$\mathrm{O}(\mathrm{C}^2 2^{2C-B}+B^2 2^{2B-A}+2^{2^{A+1}}+B2^{2C-A})$。当$\mathrm{C}=14$时可以取$\mathrm{B}=7,\mathrm{A}=3$。

false-false-true 题解

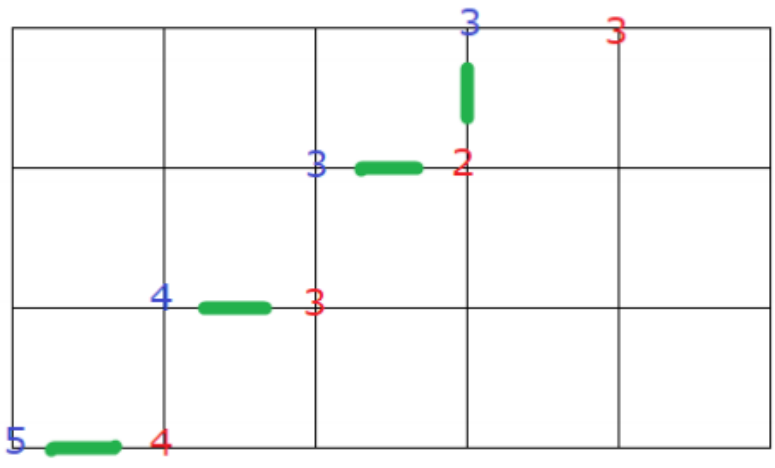
首先可以把错的题的期望转化成（n+m-对的题的数量）。

显然，当剩下的题中答案是 true 的题多于答案是 false 的题，就答 true，否则就答 false。这样就可以得到 20 分啦。

如下图如果一道题的答案是 true 就向左走，答案是 false 就向下走。题的真实的答案可能是任意一条从 s 到 t 的路径。每条路径的贡献就是这条路上所有节点的权值和。所以我们就要求随机一条 s 到 t 的路径的权值和。



我们发现每一斜行上的点在一条路径上只会走过有且仅有一次。如下图



图中每条路径都会走过一个红点和一个蓝点。所以我们可以分别统计每一斜行的值。

发现当经过绿边的路径，蓝点的值是红点的值+1，其他路径上蓝点的值等于红点的值。于是只要算出经过绿边的路径的数量。把绿边分成横着的绿边和竖着的绿边两类，然后递推一下即可。