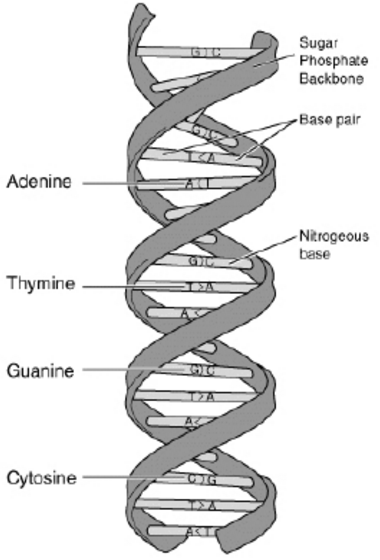
UVa1368

问题简述：给定m个长度为n的DNA序列，求一个DNA序列，使其到所有这些序列的总hamming距离尽量小，如果有多个解，输出字典顺序的最小解。

解题思路：每个DNA序列的长度相同，对每个DNA序列的每一位DNA码进行统计，选取出现次数最多的码，那么这一位的hamming距离最小。依次类推，每一位都选取出现次数最多的码，那么这样的DNA序列到n个给定的DNA序列的总hamming距离就会最小。考虑这是一个典型的可以用贪心法解决的例子。

程序中，为了计算方便，先使用map将DNA码“ACGT”映射为整数0、1、2和3，这个映射是按字典顺序的。统计计算并且得到计算结果后再将整数0、1、2和3其映射为相应的DNA码“ACGT”。这样做可以简化程序逻辑，不用写许多条件语句了。



AC的C++语言程序如下：

1. /\* UVA1368 UVALive3602 ZOJ3132 DNA Consensus String \*/
2. #include <iostream>
3. #include <map>
4. #include <cstring>
5. #define MAXN 1000
6. int dnacount[MAXN][4];
7. using namespace std;
8. int main()
9. {
10. int t, m, n, ans, i, j;
11. map<char, int> dnamap;
12. char dna[] = "ACGT", c;
13. char result[MAXN+1];
14. dnamap['A'] = 0;
15. dnamap['C'] = 1;
16. dnamap['G'] = 2;
17. dnamap['T'] = 3;
18. scanf("%d", &t);
19. while(t--) {
20. scanf("%d%d", &m, &n);
21. getchar();
22. memset(dnacount, 0, sizeof(dnacount));
23. for(i=0; i<m; i++) {
24. for(j=0; j<n; j++)
25. dnacount[j][dnamap[c = getchar()]]++;
26. getchar();
27. }
28. ans = m \* n;
29. for(i=0; i<n; i++) {
30. int maxval = dnacount[i][0];
31. int maxindex = 0;
32. for(j=1; j<4; j++)
33. if(dnacount[i][j] > maxval) {
34. maxval = dnacount[i][j];
35. maxindex = j;
36. }
37. ans -= maxval;
38. result[i] = dna[maxindex];
39. }
40. result[n] = '\0';
41. printf("%s\n", result);
42. printf("%d\n", ans);
43. }
44. return 0;
45. }