Lab10

Description

[COS 126: Assignment 10. Cracking the Genetic Code (princeton.edu)]

Solution

题目翻译一下是破解基因序列

给定一串基因序列，以及基于此序列生成的一串氨基酸序列

现在要求得出 基因的一种编码（可能会存在多种编码符合上述序列，只取最前面那种）

ps：一种氨基酸可能有多种基因序列（3个基因）编译，但一种基因序列（3个基因）仅表达一种氨基酸

单程试错法

从基因序列的第一个位置开始尝试，每三个基因决定一个氨基酸的编码，

　一旦出现某个基因编码同时表达两种氨基酸，则起始位置错误，从基因序列的下一个起始位置开始再次尝试

　氨基酸序列若是能完全表达，则输出当前编码即可

pps: eoj上此题还有氨基酸序列可能倒序表达的可能，故还要考虑氨基酸序列倒序的情况

Code

main.cpp

```
#include <stdio.h>
#include <string.h>
const int MAXN = 1e6+10;
char genecode[64];
char gene[MAXN];
char protein[MAXN];
char reprotein[MAXN];

int id(char x)
{
    switch (x)
    {
        case 'A': return 0;
        case 'C': return 1;
        case 'T': return 2;
        case 'G': return 3;
        default: return -1;
    }
}
```

```c
int hash(char *p) {
    // for (int i = 0; i < 3; i++)
    // {
    //  s *= 4;
    //  if (p[i] == 'A')
    //      ;
    //  else if (p[i] == 'C')
    //      s += 1;
    //  else if (p[i] == 'T')
    //      s += 2;
    //  else
    //      s += 3;
    // }
    int s = id(p[0]) * 16 + id(p[1]) * 4 + id(p[2]);
    return s;
    /* return a 6-bit number for the 3-character encoding p */ }

void unhash(int x) {
    char key[4];
    key[3] = '\0';
    int i = 2;
    while (i >= 0)
    {
        int k = x % 4;
        if (k == 3)
            key[i] = 'g';
        else if (k == 2)
            key[i] = 't';
        else if (k == 1)
            key[i] = 'c';
        else
            key[i] = 'a';
        x /= 4;
        i--;
    }
    printf("%s", key);
    /* print the 3-character encoding corresponding
       to the 6-bit number x */ }

int match(char *protein)
{
    int plen = strlen(protein);
    int maxpos = strlen(gene) - 3 * plen;
    int flag = 0;
    for (int pos = 0; pos <= maxpos; pos++)
    {
        flag = 1;
        for (int i = 0; i < 64; i++)
            genecode[i] = '.';

        for (int j = 0; j < plen; j++)
        {
            int key = hash(gene + pos + 3 * j);

            if (genecode[key] == '.')
                genecode[key] = protein[j];

            else if (genecode[key] == protein[j])
```

```
                continue;
            else
            {
                flag = 0;
                break;
            }
        }
        if (flag == 1)
            return pos;
    }

    return -1;
}

int main() {
    scanf("%s\n", gene);
    scanf("%s\n", protein);

    int len = strlen(protein);
    for (int i = 0; i < len; i++)
        reprotein[i] = protein[len - i - 1];
    reprotein[len] = '\0';
    /* read the contents of argv[1] into protein[0...] */
    /* read the contents of argv[2] into gene[0...] */

    /* find an encoding for protein[0...] in gene[0...]
       and record the encoding in gencode[0..63] */
    int position = match(protein);
    if (position == -1)
        position = match(reprotein);
    printf("Match found at gene position %d\n", position);

    for (int j = 0; j < 64; j++) {
        unhash(j);
        printf(" %c    ", genecode[j]);
        if ((j + 1)%4 == 0)
            printf("\n");
    }
    return 0;
}
```