

## 生信面试题

1. 写一个 python 函数, 输入一个 DNA 序列(5'->3'), 输出它的互补序列(5'->3')。

```
{python comfun, message = FALSE}
#Complement function#
def Complement(seq):
    seq = seq.upper()
    seq = seq.replace('A', 't')
    seq = seq.replace('T', 'a')
    seq = seq.replace('C', 'g')
    seq = seq.replace('G', 'c')
    return seq.upper()
#Test example#
seq = "ATCGatcgATCGatcg"
print(Complement(seq))
```

TAGCTAGCTAGCTAGC

2. 每个 SNP 位点用一个长度为 2 的 list 表示, 第一个元素为染色体编号(chr, 范围为 1~22), 第二个元素为染色体上的位置(pos)。写一个 python 函数, 输入两个正序(按 chr 和 pos 排序)的 SNP 位点 list, 输出一个合并且去重的正序 SNP 位点 list。不能使用 sorted、sort 等现成的排序函数或方法, 要求时间复杂度尽可能低。例如: Input: list1 = [[1,230], [1,4000], [2,500]], list2 = [[2,320], [6,70]] Output: [[1,230], [1,4000], [2,320], [2,500], [6,70]]

```
{python MergeSort, message = FALSE}
#MergeSort: O(nlog2n)#
def Merge_Sort(list1, list2):
    result = list()
    i = j = 0
    while i < len(list1) and j < len(list2):
        if list1[i][0] < list2[j][0]:
            result.append(list1[i])
            i += 1
        else:
            if list1[i][0] == list2[j][0]:
                if list1[i][1] < list2[j][1]:
                    result.append(list1[i])
                    i += 1
                else:
                    result.append(list2[j])
                    j += 1
            else:
                result.append(list2[j])
                j += 1
        if i == len(list1):
            result.extend(list2[j:])
        else:
            result.extend(list1[i:])
    return result
#Test example#
list1 = [[1,230], [1,4000], [2,500]]
list2 = [[2,320], [6,70]]
print(Merge_Sort(list1,list2))
```

[[1, 230], [1, 4000], [2, 320], [2, 500], [6, 70]]

3.写一个 python 函数，输入两个 DNA 序列，输出它们的最长连续公共序列。例如：

Input: sequence1 = "AGGCT", sequence2 = "GGCAT" Output: "GGC"

```
{python LCS, message = FALSE}
#LongestCommonSequence#
def LCS(s1, s2):
    m=[[0 for i in range(len(s2)+1)] for j in range(len(s1)+1)]
    mmax=0
    p=0
    for i in range(len(s1)):
        for j in range(len(s2)):
            if s1[i]==s2[j]:
                m[i+1][j+1]=m[i][j]+1
                if m[i+1][j+1]>mmax:
                    mmax=m[i+1][j+1]
                    p=i+1
    return s1[p-mmax:p]
#Test example#
sequence1 = "AGGCT"
sequence2 = "GGCAT"
print(LCS(" "+sequence1,sequence2))
```

GGC