Problem-Solving Session Rules

- Each team member must contribute to answering all the questions from the problem-session. You may lose up to 20% of your lab grade if you don't contribute.
- If a question requires you to write code, work with your teammates to write the code in this document (do not use PyCharm nor any IDE).
- Before leaving the meeting, make sure you download this document with your answers. You will probably need it later for the lab implementation.
- Check with your SLI or instructor your answers before leaving the meeting.

Do not forget to enter your name in the team members section.

Team Members

Steve Gao, 2369@rit.edu
Qiwen Quan, 5575@rit.edu

Q1

Class Node:
    __slots__ = “val”, “next”

Class DNAlist:
    __slots__ = “head”, “tail”

Strand A: head -> G -> C -> A -> None
                              ^
                              |
                            tail
Strand B: head -> C -> T -> T -> None
                              ^
                              |
                            tail

Q2
Append strand B to the end of strand A

```
def join(list1,list2):
   list1.tail.next = list2.head
   list1.tail = list2.tail
```

Q3
(a)

```
# recursive function to check if there is a match at the position cur.
Def helper(cur, s):
    If len(s) == 0:
        Return True
    If cur == None:
        Return False
    If cur.val != s[0]
        Return False
    Return helper(cur.next, s[1:])


Def find(self, s):
    Cursor = self.head
    # iterate through the whole list
    while(cursor != None):
        Found = helper(cursor, s)
        If found == True:
            Return True
        Cursor = cursor.next
    # if we reach the end of the entire list and there is still no match, return False
    Return False
```

Q3
(b)
Call find(“CT”) will return True
Substitution trace for find(“CT”) from a list corresponding to GCATT
Cursor: G helper(G, “CT”) = False
Cursor: C helper(C, “CT”) = helper(A, “T”) = False
Cursor: A helper(A, “CT”) = False
Cursor: T helper(T, “CT”) = False
Cursor: T helper(T, “CT”) = False
False

Q3
(c)
O(nk)

Q4
check insert at the front
Test case1: list_original: GCATT ind: 0 other: CT
check insert in the middle
Test case2: list_original: GCATT ind: 2 other: CT
check insert at the end
Test case3: list_original: GCATT ind: 5 other: CT
check invalid index
Test case4: list_original: GCATT ind: 7 other: CT

	List_original	Index	Other list	Expected output
Test case 1	GCATT	0	CT	CTGCATT
Test case 2	GCATT	2	CT	GCCTATT
Test case 3	GCATT	4	CT	GCATCTT
Test case 4	GCATT	7	CT	GCATT
Test case 5	GCATT	-1	CT	GCATT