## **List Problems**

Notebook: mohimac's notebook

Created: 1/26/2020 3:44 PM Updated: 1/26/2020 5:57 PM

```
# Find 2 numbers which add up to N from a list
    [1, 5, 2, 7, 6, 0,3] N - 8 result: (5,3), even if (2,6) is also valid
                                                                      Optimised Sola C(LOS=)
                                                                    Brute force: O(N"); iterate over each list index i
                      compute sum of every list[i+] + list[i]
                                                                       def find-two-nums (L, N):
                     if sum matched then return numbers
                                                                           N. sort()
Start, end = 0, len(L)-1
  det find two num (L. N):

# find 2 number from list L that add up to N
                                                                            while (start 1= end):
                                                                               cale-sum = L[start] + L[end]
      for i in range (len (L)):
                                                                                if colc-sum & N:
        for ) in range (it ), lan (L)):
                                                                                     Start += 1
             Calc.sum = Lii] + L [i]
                                                                                  elif calc-sum 7 N:
             if cole-enm on M:
                return (LCi7, LCi3)
                                                                                   else: return ( L[start], L[end])
                                                return None
Example: [1, 5, 3, 4, 6] N: 4
 i: 0 j : L calcourn: 6
i: 0 j : L calcourn: 6
return (Ltis, util)
                          (1,3)
                                                      doling a binary search
```

## **List problems**

```
PESCY
    Modify a list based on some conditions
     Example: nemove all numbers with a particular value from a list
OPTION 1: Use list comprehension
                                                                              EXAMPLE: List: [1,2,3,4,3] velue: 3
      result-list = [elem for elems in list if put-condition-here]
                                                                                1.0 [1,2,3,4,3]
      This requires extra-space
                                                                                1:1 [1, 2, 3, 4, 3]
      To solve this in-place use the below option
                                                                                 1.2 [1,2,9,4,5] -> POP(2)
OPTION 2: Iterate the list and pop out every element tenat matches the condition.
                                                                                 is @ [1, 2, w, 3] So not in cramer when element
                                                                                 1:3 [1, 2, 4, 3] -> POP(3)
      def vemove-element-from-list ( L, val):
                                                                                  1:3 [1,2,4] - Loopends as its len(list)
                 i, n = 0, len(L)
                 while i in range (n)
                     if L[i] == V81:
                                          # do not increment i because everytime
                         L.pop(i)
                                              element is popped the next elements
                           n --1
                                             are left-shifted so the index will
                      else:
                                              hore a different element that needs
                                                 to be checked.
                   neturn L
```

## **List problems**

```
# Merge 2 sorted lists
      [1, 2, 3, 4], [4, 5, 6, 7] => [1, 2, 3, 4, 4, 5, 6, 7]
Algo: 1. Ptr i to maintain curr position in lists
                                                                                      3. At each iteration compare lists[i] to lists[i]. Add min(littl[i], lists[i]) to result list
     4. At the end of Herotion look for leftover element in the longer list and odd them.
                                                                                          in the militarial res [2, 5, 10, 12] ellis
       i, j = 0.0

for idx in range(n):

if let1[i] > 10.1
                                                                                          inia u: [1,2,20] 12: [2, 5, 10, 12] R: [1,2,2]
 def merge-lists (1st1, 1st2):
                                                                                          1:27:1 Li: [1,2,5] Li: [2,5,10,12] &: [1,1,2,5]
                                                                                           # iterate for min length of 2 Usts
           if let 1 [1] > 1 (cf 2 (5)) : # mainfoin pointers of both the links
                                                                                             since the iteration has anded one know that at least one of the lists
                                                                                            1:2):4 so loop ands here
                                                                                             has been fully processed (remarks, while (Lim(L1) DND ) (Len(L2))
                  j + + 1
             else: result. append (1st2[i])
                                                                                           =7 Novo check if either of the late has any left-over element.
                                               # Look for left over elements in the
                    1+=1
                                                                                             This is possible only if ptn ( len(list)
                                                    list that has longer lawth
                                                                                             If yes then append leftorer element: list[ptr:]
          if ic leachers:
                for item in 1sts[is]:
                                                                                            i < len(LI) so append elements from LITII to result
                     result. append (item)
           if ) < len(1s+2):
                                                                                                 L1[2:]
                  for item in 1st2[j:]:
                                                                                                   4 result . [1, 2, 2, 5, 10, 12, 20]
                        result. append (item).
```