```
Here are the commands I used to set up the github and export environment.

echo "# HW1" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/ChenweWu/HW1.git

git push -u origin main

conda create --name medai

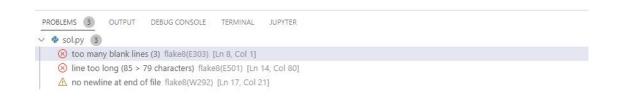
conda activate medai

conda env export > environment.yml
```

```
PS C:\Users\chenw\cs197\hw1> 1s
    Directory: C:\Users\chenw\cs197\hw1
Mode
                    LastWriteTime
                                          Length Name
            9/15/2022 10:44 PM
9/15/2022 10:38 PM
-a----
                                               80 environment.yml
-a----
                                               30 README.md
PS C:\Users\chenw\cs197\hw1> git add .\environment.yml
PS C:\Users\chenw\cs197\hw1> git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
```

Using Lint helped me correct up a lot of bad coding styles such as extra whitespaces and long lines.

```
① README.md
             sol.py > 😝 Solution
      class ListNode(object):
         def __init__(self, val=0, next=None):
             self.val = val
  4
            self.next = next
      class Solution(object):
  8
          def mergeKLists(self, lists):
  9
 10
 11
              :type lists: List[ListNode]
              :rtype: ListNode
 12
 13
 14
              CS197 HW1 Q2. LC https://leetcode.com/problems/merge-k-sorted-lists/solution/
 15
 16
 17
              return lists
```



It also helps me with minor issues like reminding me of variables created but not used, and using != None versus using is not

I

```
None.

class Solution(object):

def mergeKLists(self, lists):

"""

:type lists: List[ListNode]

:rtype: ListNode

CS197 HW1 Q2.

LC https://leetcode.com/problems/merge-k-sorted-lists/solution/

brute force iteration

"""

all_list_val = []

for list in lists:

while list.next != None:

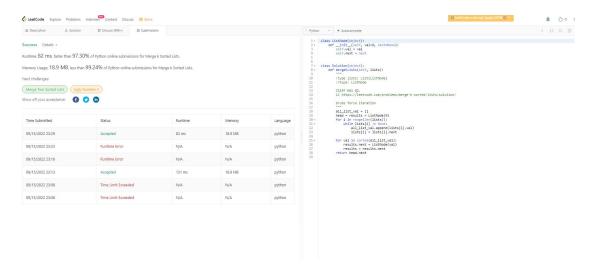
all_list_val.append(list.val)

results = ListNode()

return lists
```

I am going to use a brute force iteration method that suprisingly beat 97% of the solutions. At first the code wasn't even runnable, but with the help of debugger and keeping track of my variables, I was able to get my solution accepted.

```
class Solution(object):
     def mergeKLists(self, lists):
         :type lists: List[ListNode]
         :rtype: ListNode
         CS197 HW1 Q2.
         LC https://leetcode.com/problems/merge-k-sorted-lists/solution/
         brute force iteration
         all_list_val = []
         for list in lists:
             while list.next is not None:
                 all list val.append(list.val)
         results = ListNode()
         for val in all_list_val:
             results.next = ListNode(val)
             results = results.next
         return lists
```



Here is the github link: https://github.com/ChenweWu/HW1