def uniques(string): #with data structures O(n)

stores=dict()

for ith in string:

if stores.get(ith,0) !=0:

return False

stores[ith] = 1

return True

print(uniques("HELLO"))

def uniques\_pro(string): #without data structures O(n\*logn +n)= o(n\*logn)

string = sorted(string)

for ith in range(len(string)-1):

if string[ith+1]==string[ith]:

return False

return True

print(uniques\_pro("HELLO"))

Question to JP, should we look to the answer after we implemented a solution, or should we first discuss it with each other to optimize the current code, i’d say that from all the questions that I resolve in a day eachother should pick one and implement a better solution, i.e. improve the space complexity.

Id say that we should make some questions before actually start coding and from all the questions code a solution we’d assume that they answered. I.e in this problem i assume that the questions that I would have asked are could be any char or ASCII?[a:Any char] with any other data structure you mean only use the string and do not declare any other strings or arrays, or other[A:only use string], Can I use sort functions for python[A:yes you can use sorted]

So we clarify the questions we assumed for this problem cause we should be always asking questions. Im not suggesting alwasy pick the harder “answers” but trying to answer the “easy” answers and one of the harder ones would be interesting like contemplating 2 examples