HW5: Backtracking + Greedy Algorithm

Question 1:

a. describe backtracking problem

function amount _ helper (A, S, start, array):

if (A == =0): 11 base case

return []

for i in vange (length of A):

append. A [i] 11 Add to 115+

A[i] - A [odd] > target "if nist > than target

POP. ACI] 11 remove that item AKA backtracking

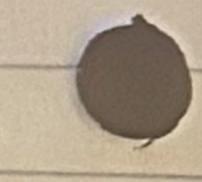
return rusult

function Amount (A,5):

amount_helper(A,S,O,[])

Meturn final- wruit

* my way of solving requires a helper. It will go through the list, append items unless goes over target sum. if it equals o then it will teturn that list,



b. in gradercope

C. The time complexity is $O(2^n)$ = exponential. Backtracking means going through the whole list + returning all results that equal target or what you'll solving. For the example above, it would have to check the list we all of the ways it could satisfy. So this is a, can be, slow as it can exponentially grow.

Question 2:

 α .

def feed Dog (nunger-level, biscuit-size)

for i in vange (nunger_1evel):

tor i in vange (biscuit-size):

if (biscuit 2 hunger)

COUNT + -1
biscuited nunger [i]

through both list then if thek are more biscuits than hunger, add to count and remove that biscuit and countinue

For the greedy aspect, I can sour in also order to feed the most

144UVN COUNT

HW 5: avestion 2 C. time complexity The time complexity of this problem is o(n2). This is are to the fact that I will have two loops iterating, but it's just the Iten9th of both 115ts, so its Q(n2)