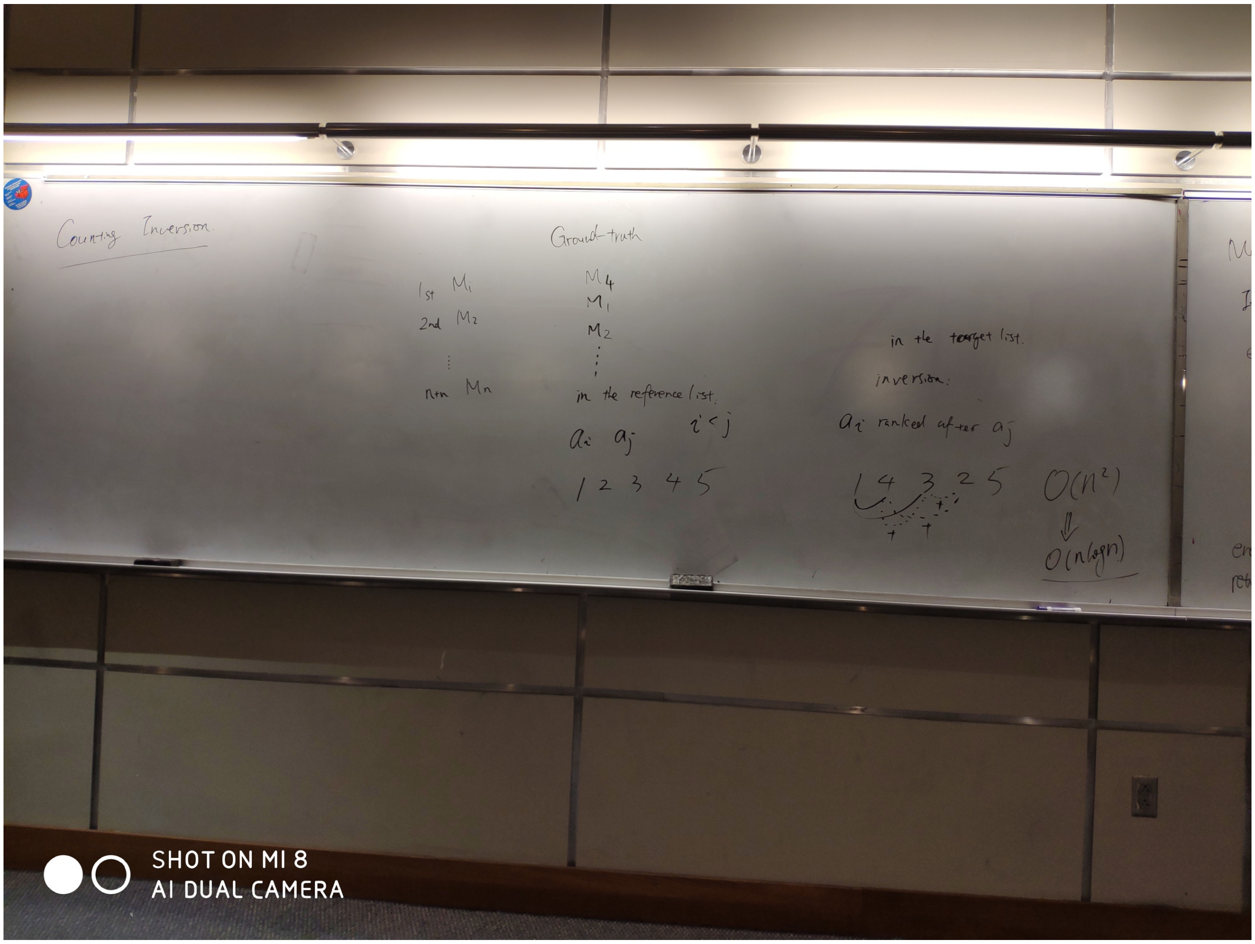
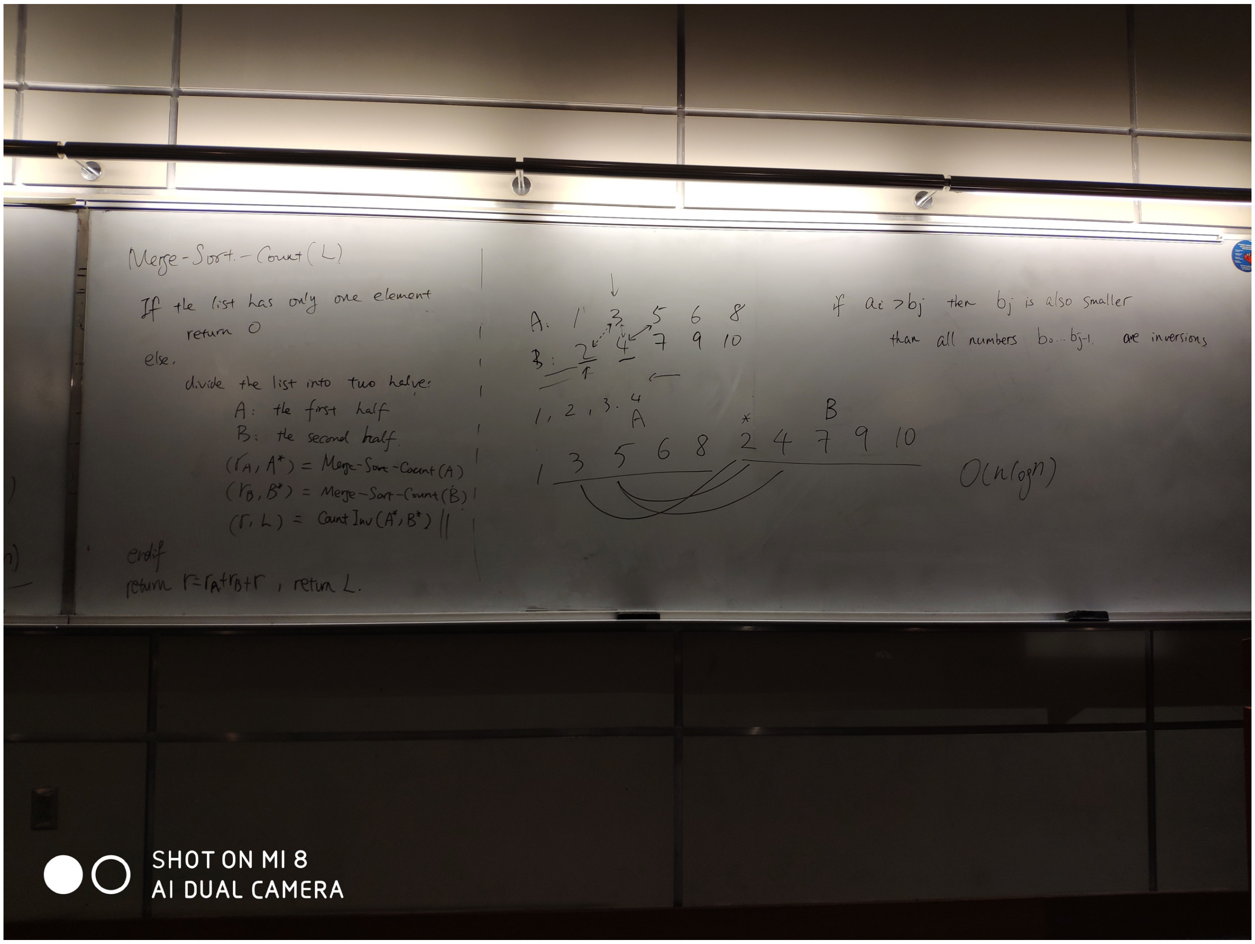
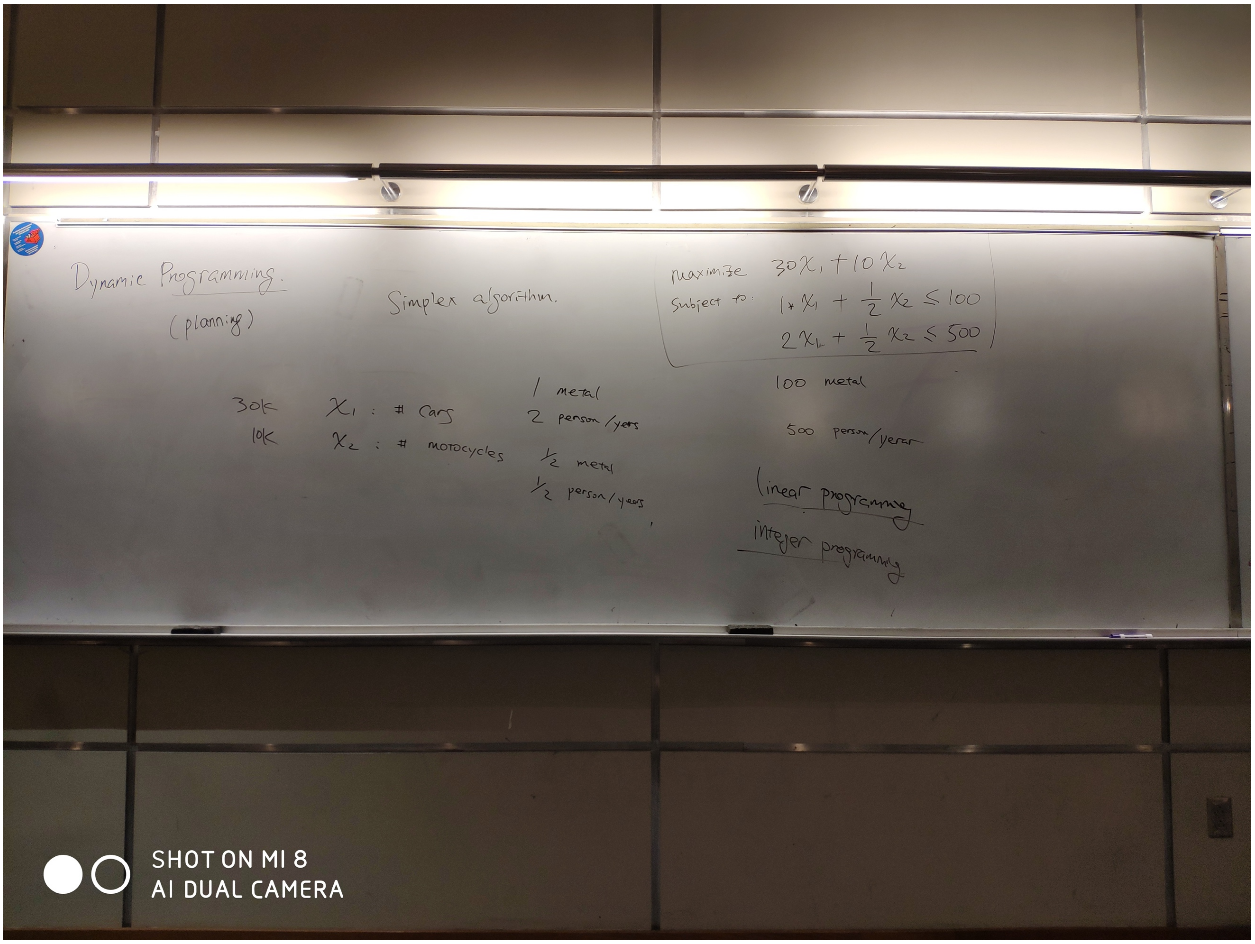


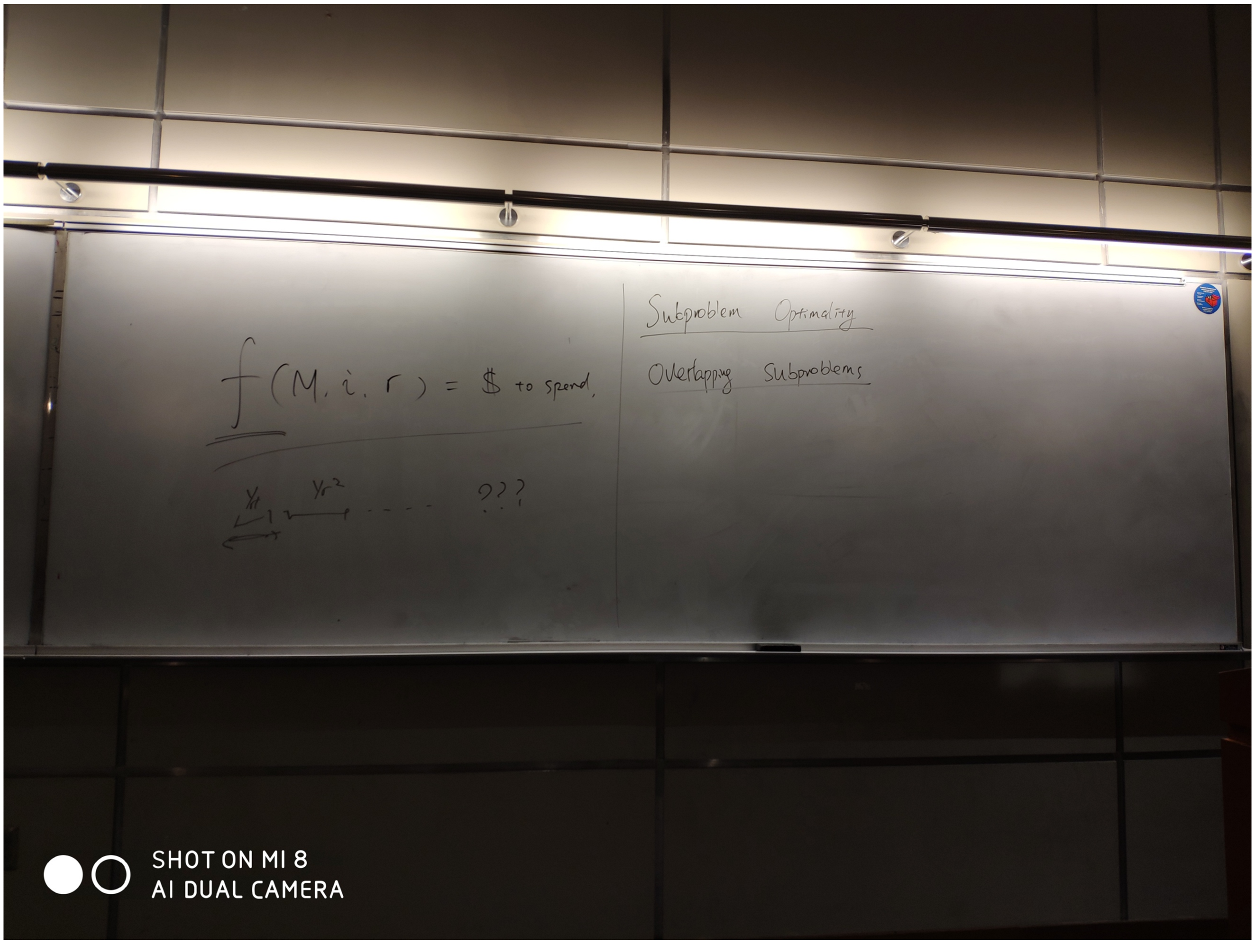
Counting Inversion
1st M1
2nd M2
nth Mn
Ground-truth
M4
M1
M2
in the reference list.
ai aj
i < j
1 2 3 4 5
in the target list.
inversion:
ai ranked after aj
1 4 3 2 5
O(n^2)
O(nlogn)
SHOT ON MI 8
AI DUAL CAMERA

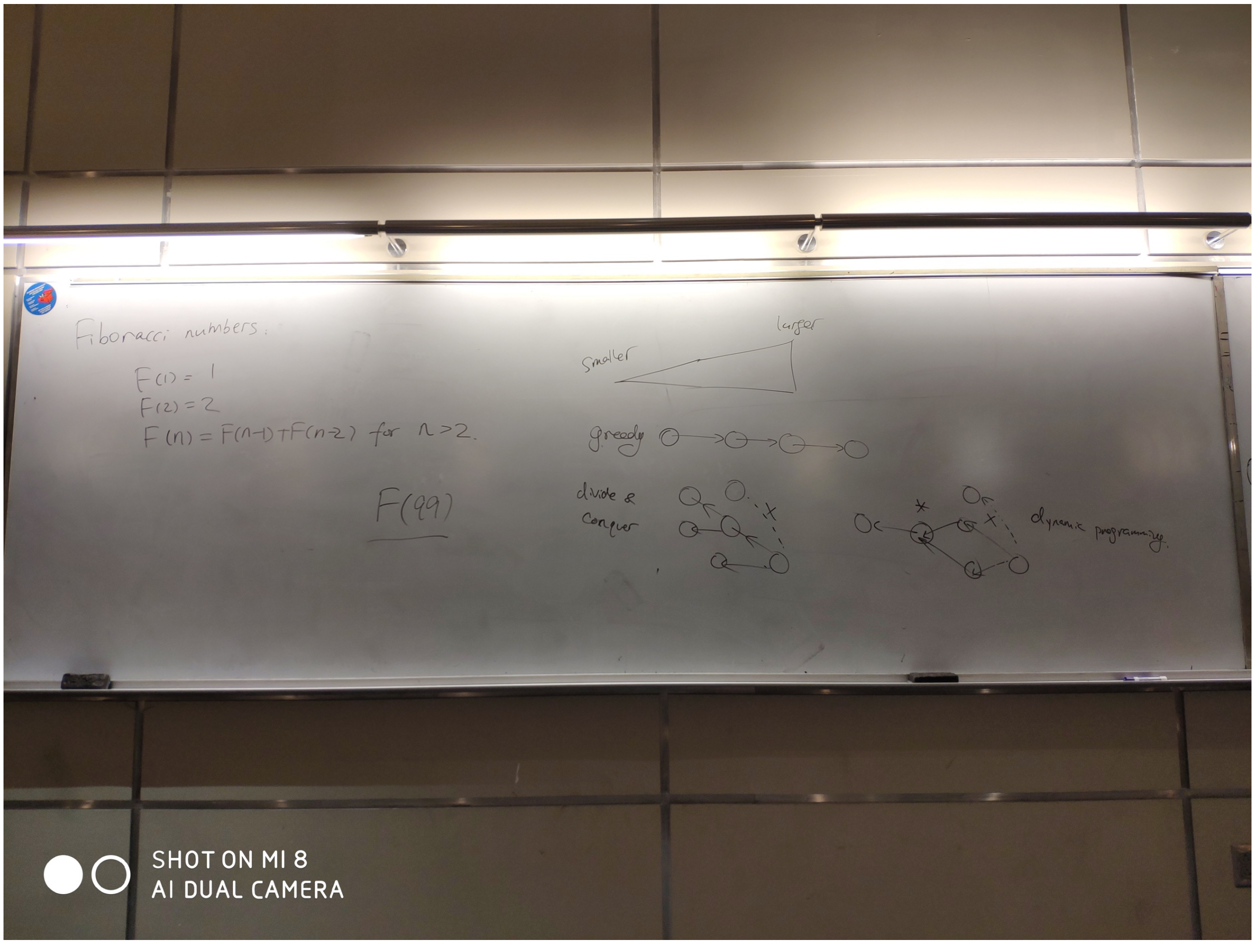


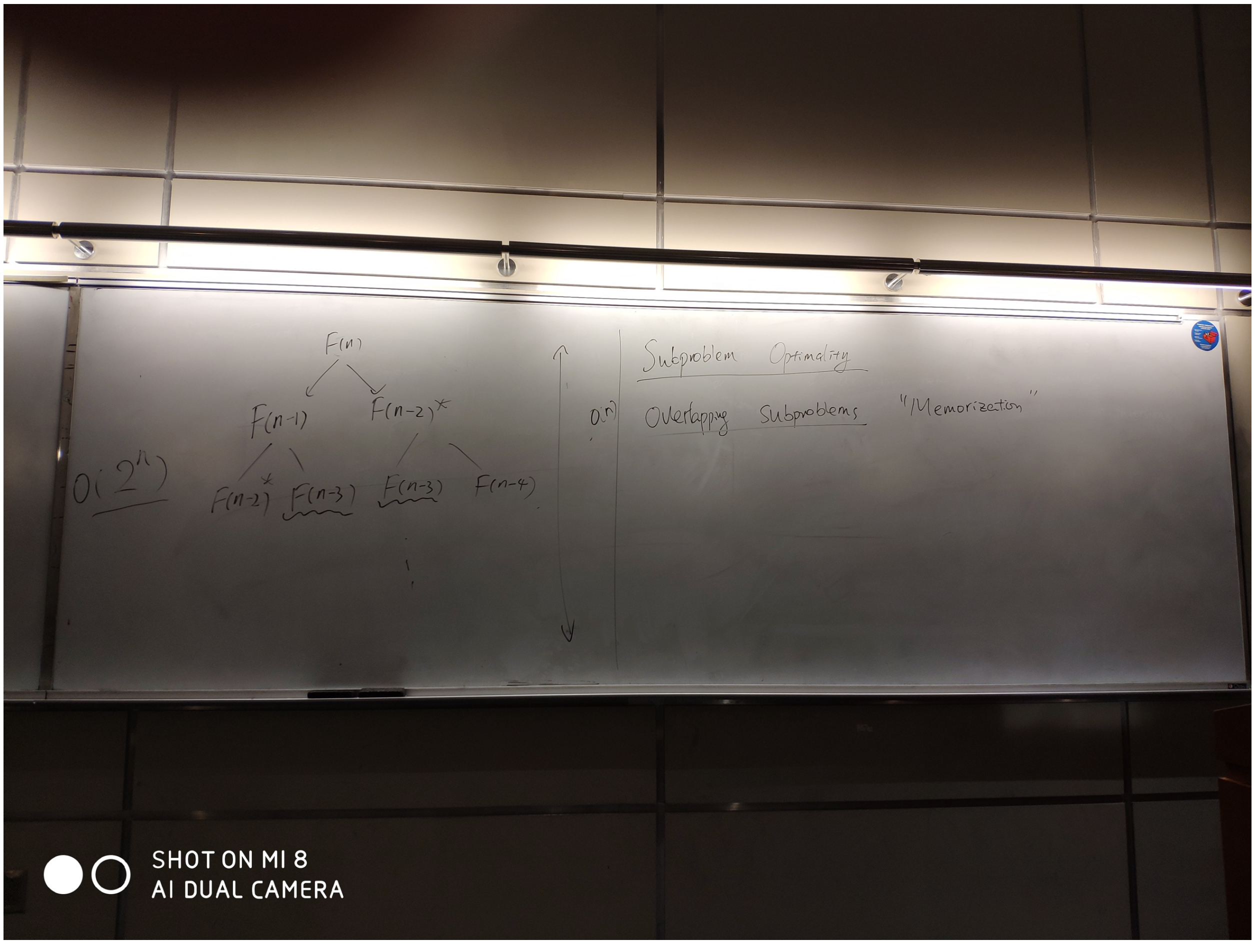
Merge-Sort-Count(L)
If the list has only one element
return 0
else,
divide the list into two halves
A: the first half
B: the second half
(r_A, A^*) = Merge-Sort-Count(A)
(r_B, B^*) = Merge-Sort-Count(B)
(r, L) = Count Inv(A^*, B^*)
endif
return $r = r_A + r_B + r$, return L.
A: 1 3 5 6 8
B: 2 4 7 9 10
1, 2, 3, 4
A
B
1 3 5 6 8 2 4 7 9 10
if $a_i > b_j$ then b_j is also smaller
than all numbers $b_0 \ldots b_{j-1}$ are inversions
$O(n \log n)$
SHOT ON MI 8
AI DUAL CAMERA



Dynamic Programming.
(planning)
Simplex algorithm.
maximize $30x_1 + 10x_2$
Subject to: $1 * x_1 + \frac{1}{2} x_2 \le 100$
$2x_1 + \frac{1}{2} x_2 \le 500$
30k x_1 : # cars
10k x_2 : # motocycles
1 metal
2 person/years
½ metal
½ person/years
100 metal
500 person/yerar
linear programming
integer programming
SHOT ON MI 8
AI DUAL CAMERA


f(M, i, r) = $ to spend,
???
Subproblem Optimality
Overlapping Subproblems
SHOT ON MI 8
AI DUAL CAMERA


Fibonacci numbers:
F(1) = 1
F(2) = 2
F(n) = F(n-1)+F(n-2) for n>2.
F(99)
smaller
larger
greedy
divide &
conquer
dynamic programming.
SHOT ON MI 8
AI DUAL CAMERA


F(n)
F(n-1)
F(n-2)*
F(n-2)*
F(n-3)
F(n-3)
F(n-4)
O(2^n)
Subproblem Optimality
O(n)
Overlapping Subproblems
"Memorization"
SHOT ON MI 8
AI DUAL CAMERA