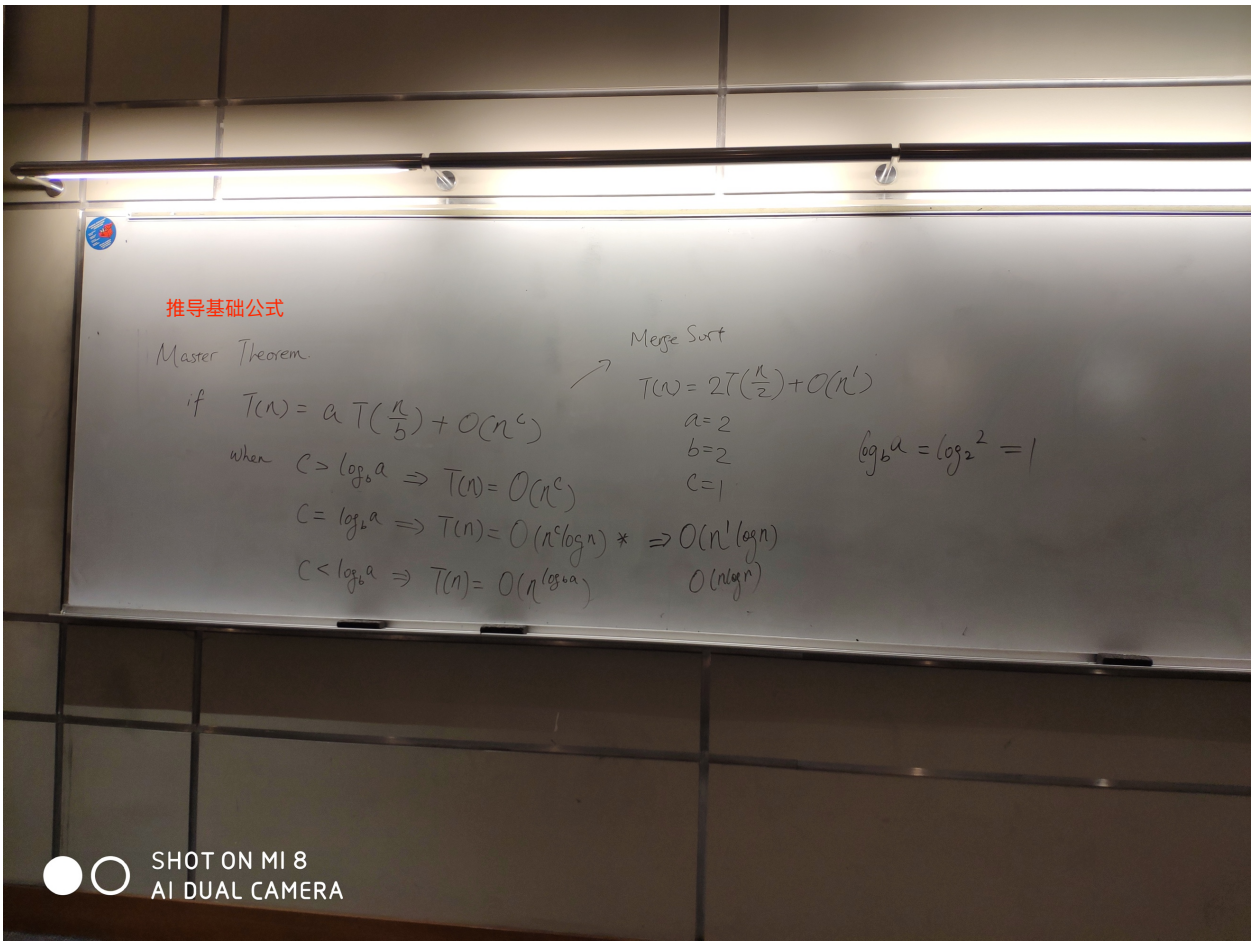
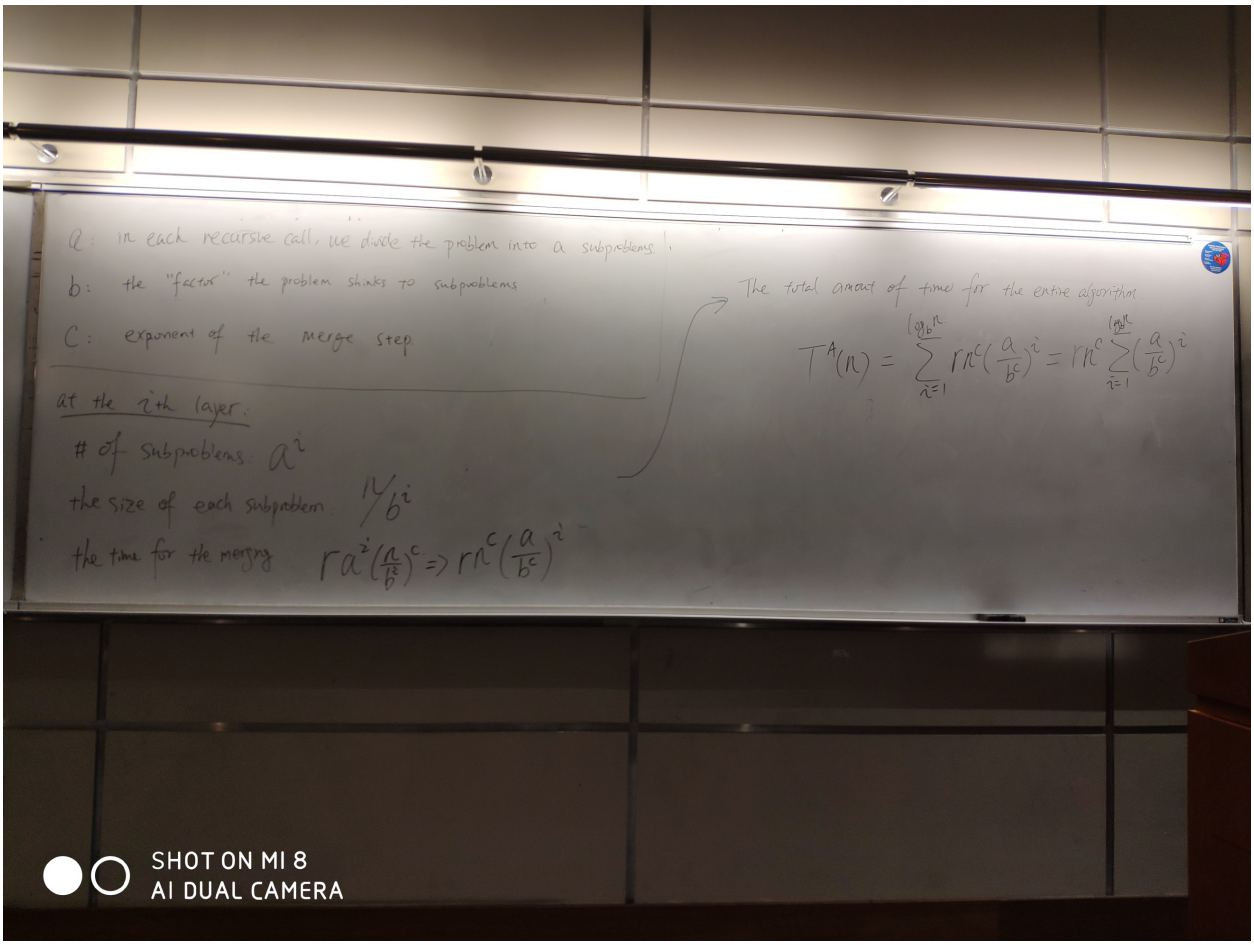
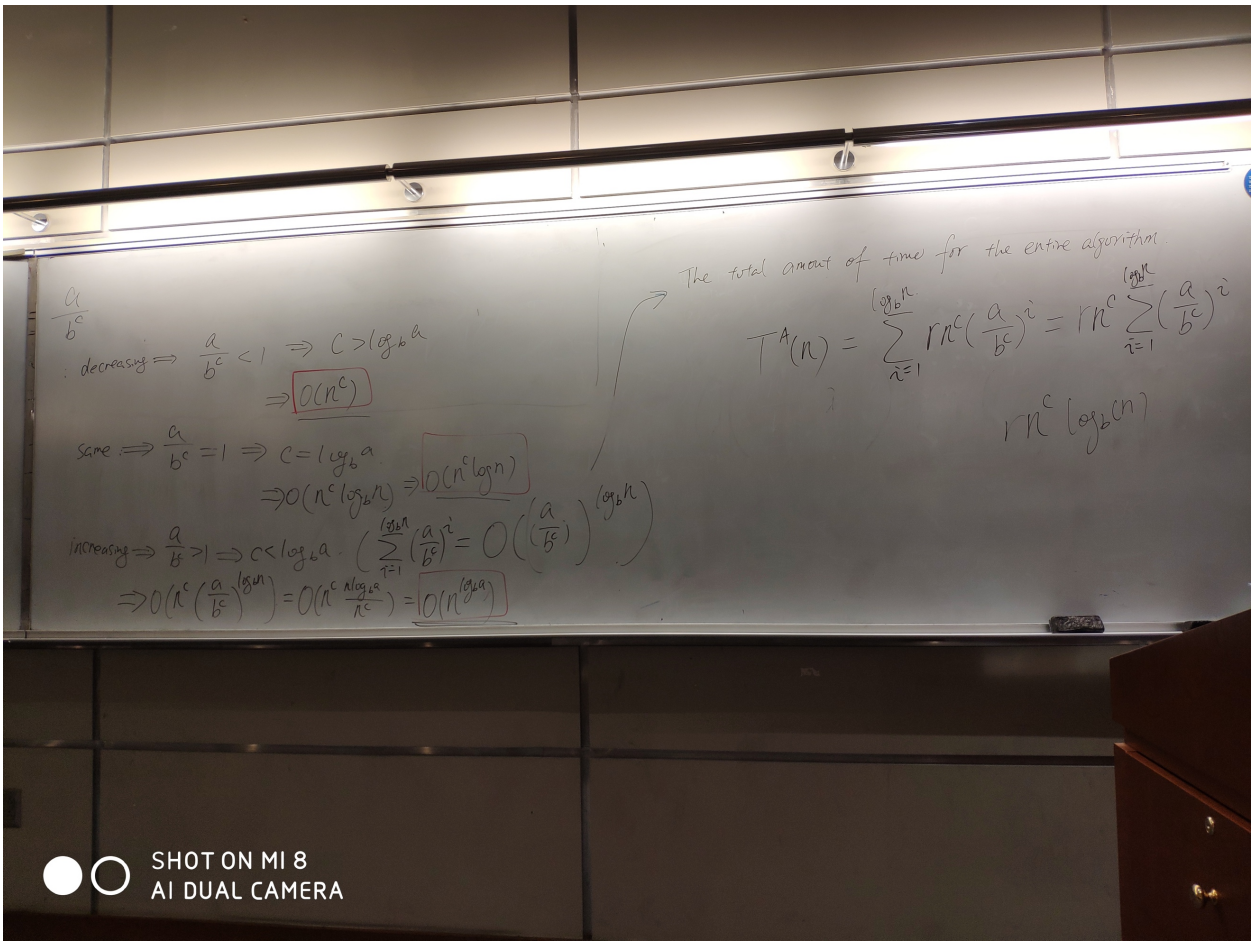
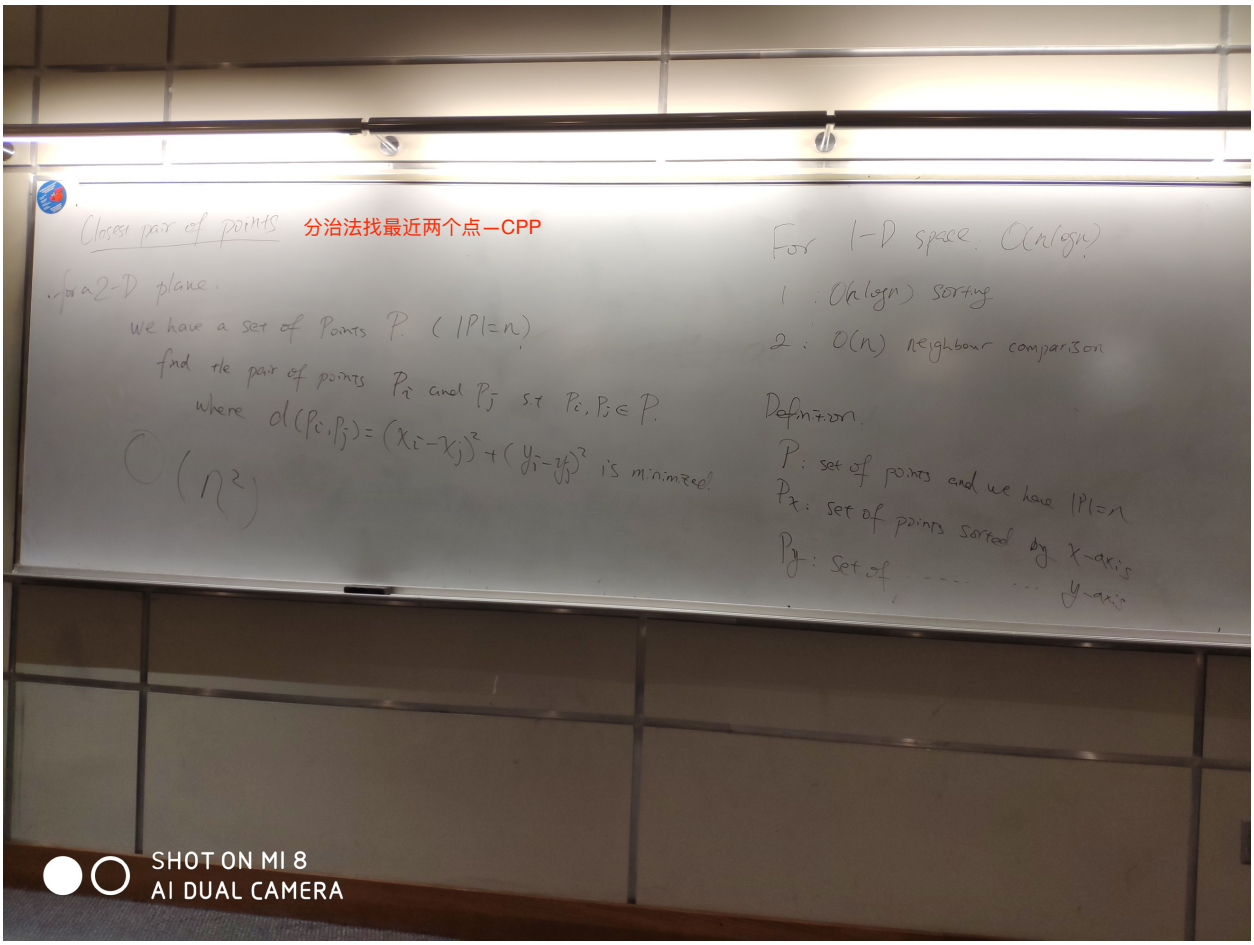
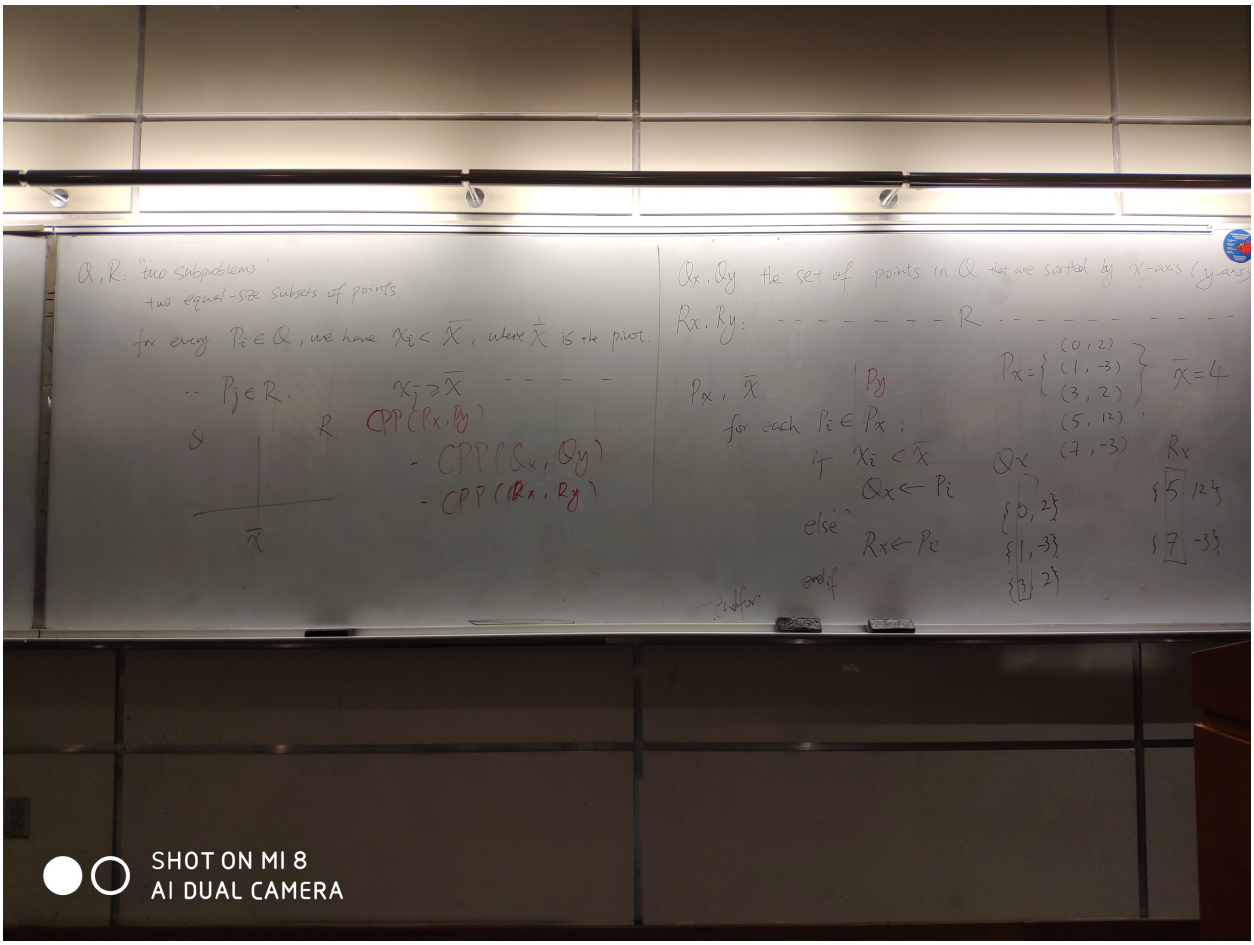
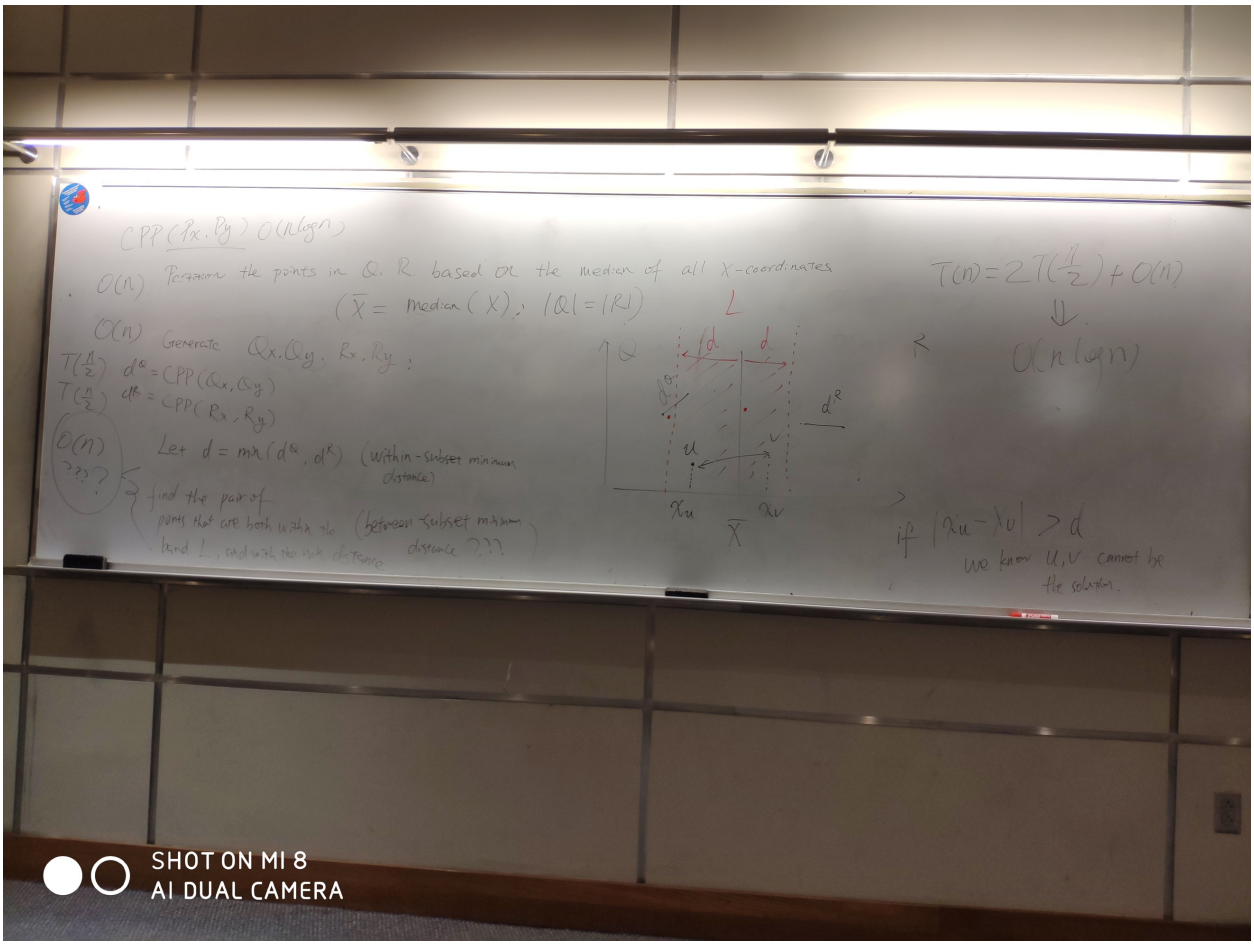

推导基础公式
Master Theorem
if $T(n) = aT(\frac{n}{b}) + O(n^c)$
when $c > \log_b a \Rightarrow T(n) = O(n^c)$
$c = \log_b a \Rightarrow T(n) = O(n^c \log n)$ *
$c < \log_b a \Rightarrow T(n) = O(n^{\log_b a})$
Merge Sort
$T(n) = 2T(\frac{n}{2}) + O(n^1)$
$a = 2$
$b = 2$
$c = 1$
$\log_b a = \log_2 2 = 1$
$\Rightarrow O(n^1 \log n)$
$O(n \log n)$
SHOT ON MI 8
AI DUAL CAMERA

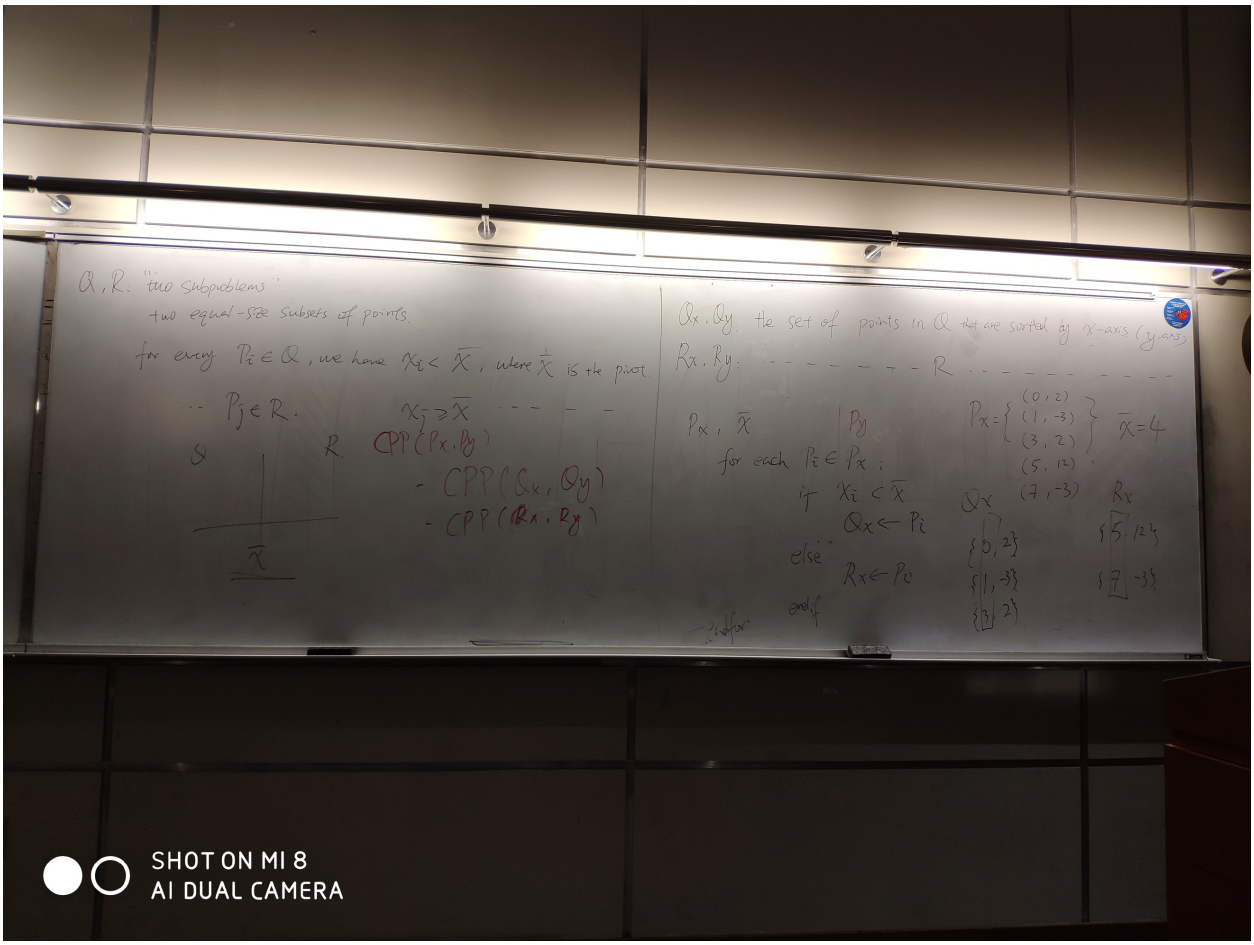

a: in each recursive call, we divide the problem into a subproblems.
b: the "factor" the problem shinks to subproblems
c: exponent of the merge step
at the ith layer:
# of subproblems: a^i
the size of each subproblem: n/b^i
the time for the merging: $ra^i(\frac{n}{b^i})^c \Rightarrow rn^c(\frac{a}{b^c})^i$
The total amount of time for the entire algorithm.
$T^A(n) = \sum_{i=1}^{\log_b n} rn^c(\frac{a}{b^c})^i = rn^c \sum_{i=1}^{\log_b n}(\frac{a}{b^c})^i$
SHOT ON MI 8
AI DUAL CAMERA


$\frac{a}{b^c}$
decreasing $\Rightarrow \frac{a}{b^c} < 1 \Rightarrow c > \log_b a$
$\Rightarrow O(n^c)$
same $\Rightarrow \frac{a}{b^c} = 1 \Rightarrow c = \log_b a$
$\Rightarrow O(n^c \log_b n) \Rightarrow O(n^c \log n)$
increasing $\Rightarrow \frac{a}{b^c} > 1 \Rightarrow c < \log_b a$
$\Rightarrow O\left(n^c \left(\frac{a}{b^c}\right)^{\log_b n}\right) = O\left(n^c \frac{n^{\log_b a}}{n^c}\right) = O(n^{\log_b a})$
The total amount of time for the entire algorithm
$T^A(n) = \sum_{i=1}^{\log_b n} r n^c \left(\frac{a}{b^c}\right)^i = r n^c \sum_{i=1}^{\log_b n} \left(\frac{a}{b^c}\right)^i$
$r n^c (\log_b n)$
SHOT ON MI 8
AI DUAL CAMERA


Closest pair of points 分治法找最近两个点—CPP
For 1-D space: O(nlogn)
1: O(nlogn) sorting
2: O(n) neighbour comparison
in a 2-D plane:
we have a set of Points P. (|P|=n)
find the pair of points Pi and Pj s.t Pi, Pj ∈ P.
where d(Pi,Pj) = (xi-xj)^2 + (yi-yj)^2 is minimized
O(n^2)
Definition:
P: set of points and we have |P|=n
Px: set of points sorted by x-axis
Py: set of ---- ... y-axis
SHOT ON MI 8
AI DUAL CAMERA


Q, R: "two subproblems"
two equal-size subsets of points.
for every Pi ∈ Q, we have xi < x̄, where x̄ is the pivot.
Pj ∈ R.
xj ≥ x̄
S
R
x̄
CPP(Px, Py)
- CPP(Qx, Qy)
- CPP(Rx, Ry)
Qx, Qy: the set of points in Q that are sorted by x-axis (y-axis)
Rx, Ry: R
Px, x̄
Py
for each Pi ∈ Px :
if xi < x̄
Qx ← Pi
else
Rx ← Pi
end if
end for
Px = {(0, 2), (1, -3), (3, 2), (5, 12), (7, -3)}
x̄ = 4
Qx
{0, 2}
{1, -3}
{3, 2}
Rx
{5, 12}
{7, -3}


CPP(Px, Py) O(nlogn)
O(n) Partition the points in Q, R based on the median of all X-coordinates
($\bar{X}$ = median(X), |Q| = |R|)
O(n) Generate Qx, Qy, Rx, Ry :
$T(\frac{n}{2})$ d^Q = CPP(Qx, Qy)
$T(\frac{n}{2})$ d^R = CPP(Rx, Ry)
Let $d = \min(d^Q, d^R)$ (within-subset minimum distance)
O(n) ??? find the pair of points that are both within the band L, and with the min distance (between-subset minimum distance ???)
$T(n) = 2T(\frac{n}{2}) + O(n)$
$\Downarrow$
$O(n \log n)$
L
Q
R
d
d^Q
d^R
u
v
x_u
x_v
$\bar{X}$
if $|x_u - x_v| > d$
we know u, v cannot be the solution.
SHOT ON MI 8
AI DUAL CAMERA


Q, R: "two subproblems"
two equal-size subsets of points.
for every $P_i \in Q$, we have $x_i < \bar{x}$, where $\bar{x}$ is the pivot.
$P_j \in R$. $x_j \geq \bar{x}$
S
R
$\bar{x}$
CPP(P_x, P_y)
- CPP(S_x, Q_y)
- CPP(R_x, R_y)
Q_x, Q_y: the set of points in Q that are sorted by x-axis (y-axis)
R_x, R_y: ---------- R ----------
P_x, $\bar{x}$ P_y
for each $P_i \in P_x$:
if $x_i < \bar{x}$
$Q_x \leftarrow P_i$
else
$R_x \leftarrow P_i$
end if
P_x = {(0, 2), (1, -3), (3, 2), (5, 12), (7, -3)}
$\bar{x} = 4$
Q_x
{0, 2}
{1, -3}
{3, 2}
R_x
{5, 12}
{7, -3}