1.1 closest - Pair (P)

$\Theta(1)$ if $|P| \le 3$ brute-force finding closest and return.

find middle that can divide the surface into two parts that contain the same number of points

$2T(\frac{n}{2})$ left = closest - Pair (left part)

right = closest - Pair (right part)

minn = min (left , right).

find (left - middle) points $p_0 \cdots p_i$

$\Theta(n)$ for point p_i

find $p_0 \cdots p_k$ next the right and height

between (heightp_i + minn , height p_i - minn)

compute p_i with p_k.

update the closest distance.

return the closest distance.

main (P)

// the original status is different because it's a circle.

$\Theta(n \log n)$ P = sort (P) by angle ($-180°$, $180°$)

$T(\frac{n}{2})$ left = closest - Pair (P - left)

$T(\frac{n}{2})$ right = closest - Pair (P - right).

minn = min (left . right).

$\Theta(n)$ find $(-minn, minn)$ points $p_0 \cdots p_i$

for point p_i

find $p_0 \cdots p_k$ next the right and height

between (heightp_i + minn, height p_i − minn)

compute p_i with p_k.

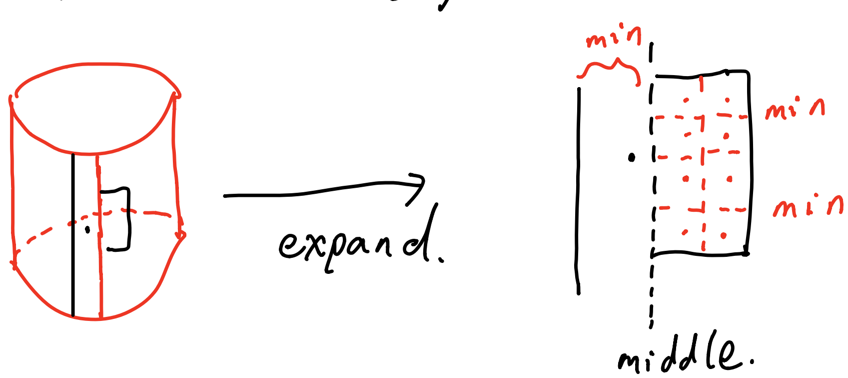
update the closest distance.

$\Theta(1)$. return min (left, right, closest distance)

1.2

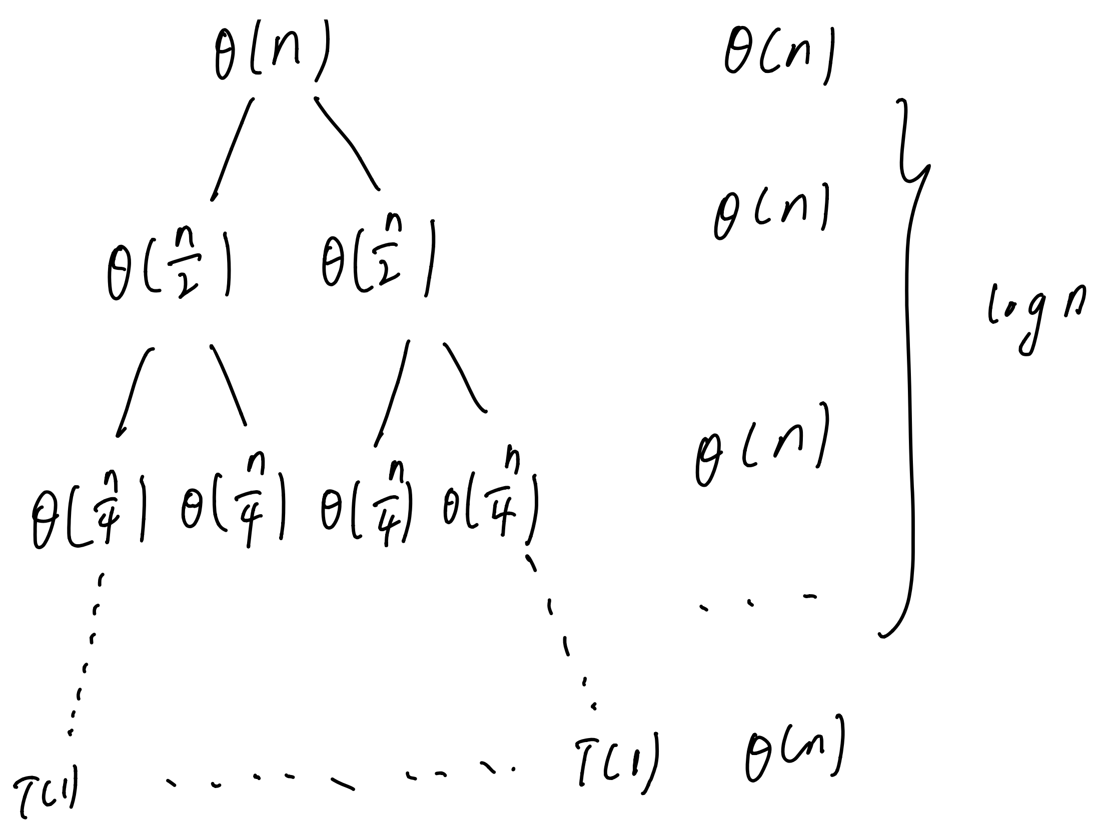
$$T(n) = \begin{cases} \Theta(1) & \text{if } n<3 \\ 2T(\frac{n}{2}) + \Theta(n) & \text{if } n>3 \end{cases}$$

$$T(n) = \Theta(n \log n) = O(n \log n)$$



less than 8

1.3. $T(n) = 2T(\frac{n}{2}) + \Theta(n)$



$$T(n) \le O(n \log n)$$