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 $(-180°, 180°)$

$T(n)$ minn = closeset - Pair (P)

(... (with the original line)

(not include original ...).

$\Theta(n)$ 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.

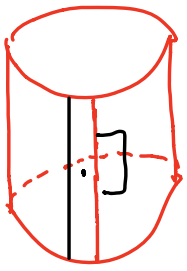
update the closest distance.

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

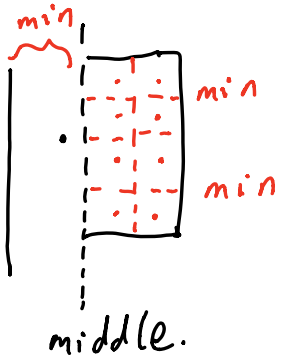
1.2

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

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

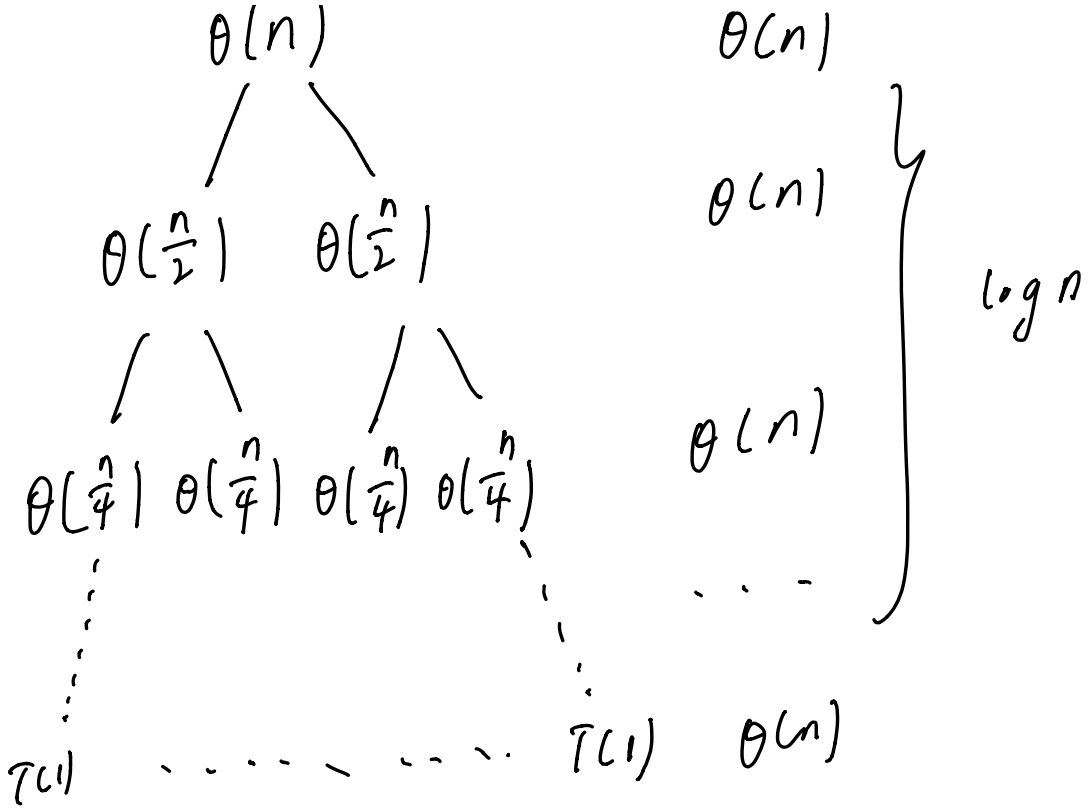


expand.



less than 8

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



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