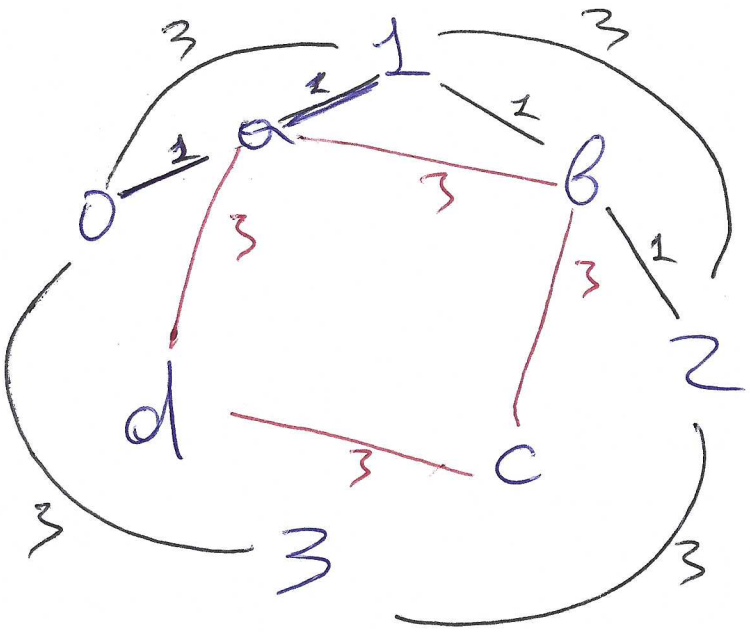
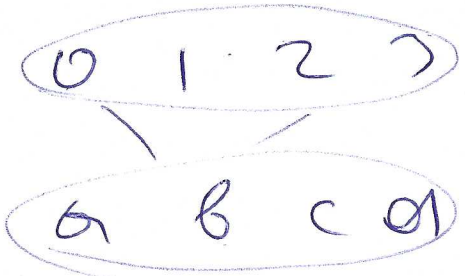
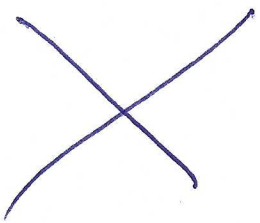
TSP using divide and conq



split



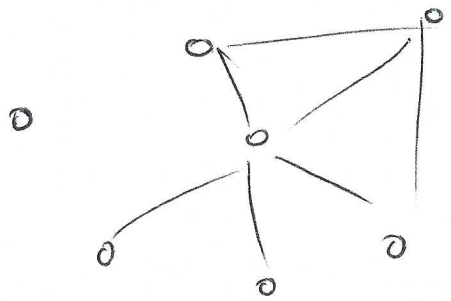
combine!??



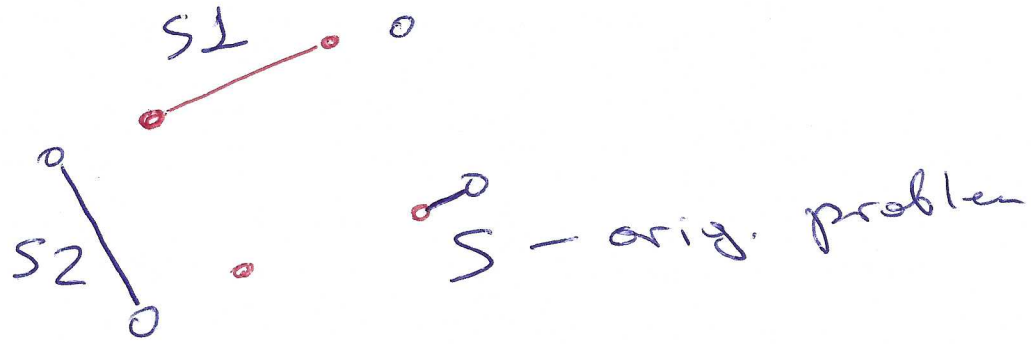
0-1 Knapsack

---

closest pair

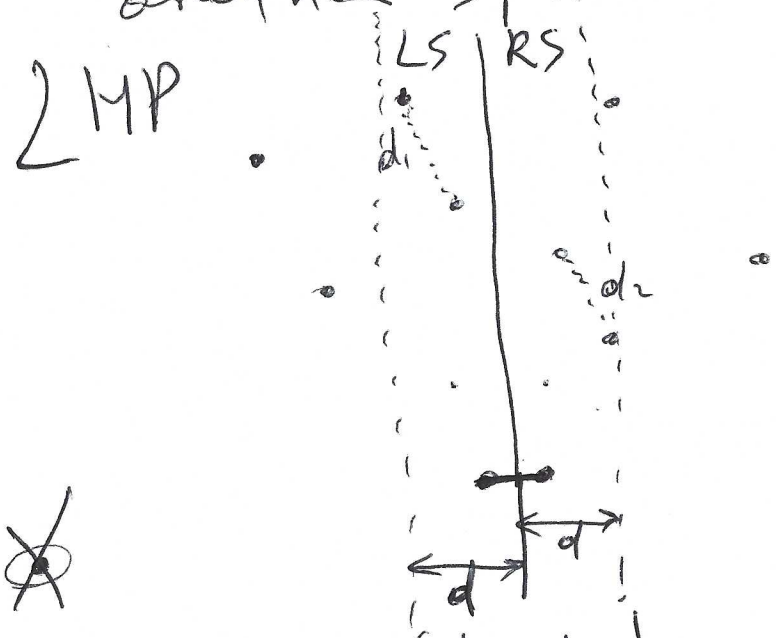


X



another split!

LMP



$d = \min(d_1, d_2)$

RMP

$T(n) = 2T(\frac{n}{2}) + \underbrace{\frac{n^2}{4}}_{\text{comb.}}$

size(LS) * size(RS)

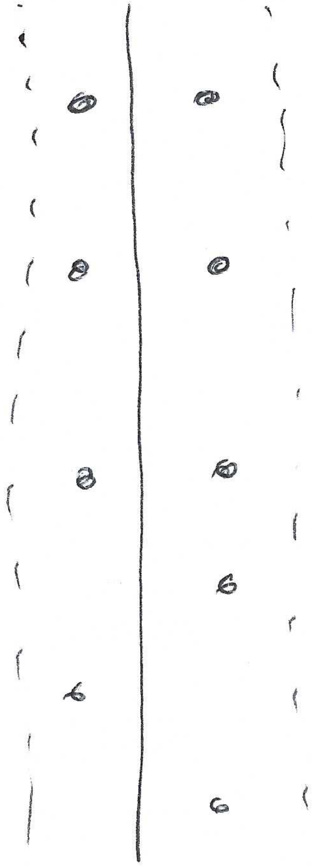
$\frac{n}{2}$ $\frac{n}{2}$

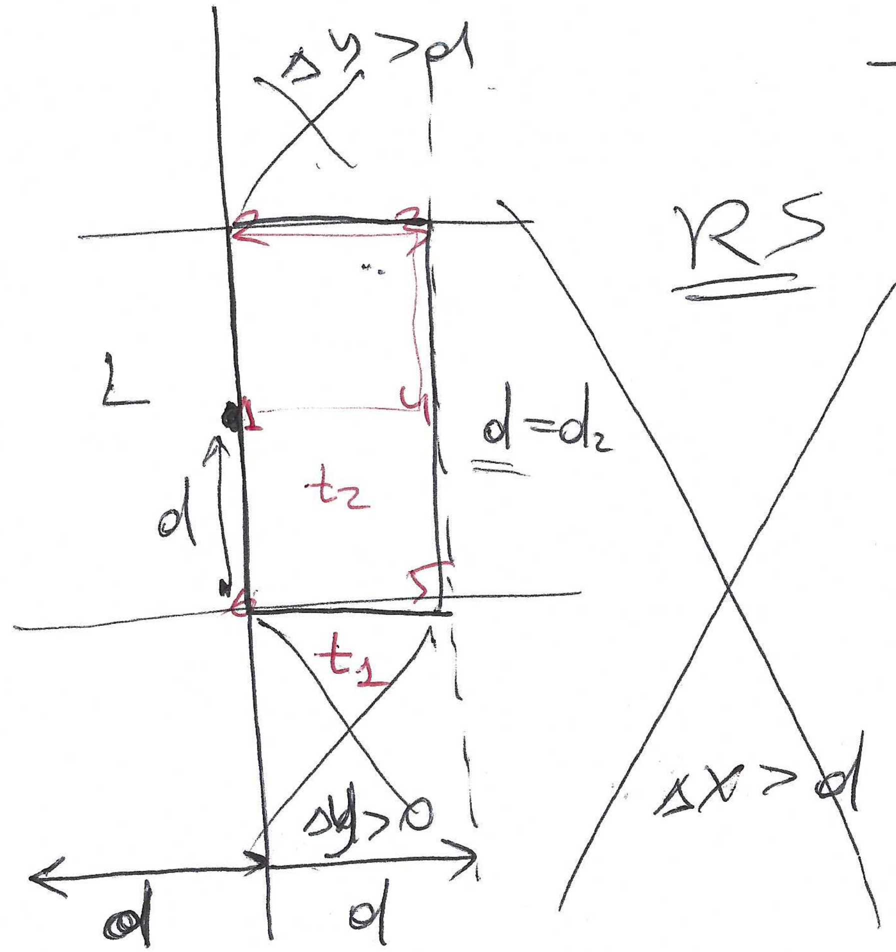
$O(n^2)$

X with p.3 optimization

$$\begin{cases} T(n) = 2T(\frac{n}{2}) + \frac{n}{2} \times 6 \dots \\ T(1) = 1 \end{cases}$$

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





RS

Δ

$t_2 > t_1$ t_1

if(Δy > 0) {skip}

else {

t_2 new $d^2 = \Delta x^2 + \Delta y^2$

1 single sort → preset → y —

Split:

1) $x_{split} = \frac{x_{min} + x_{max}}{2}$? $O(n)$ | $O(1)$

: I..... : $O(n)$

$= \frac{\sum x_i}{n}$ (avg) $O(n)$ | $O(n)$

$O(n \log n)$? | $O(1)$

$x_{split} = x_{50\%}$

---

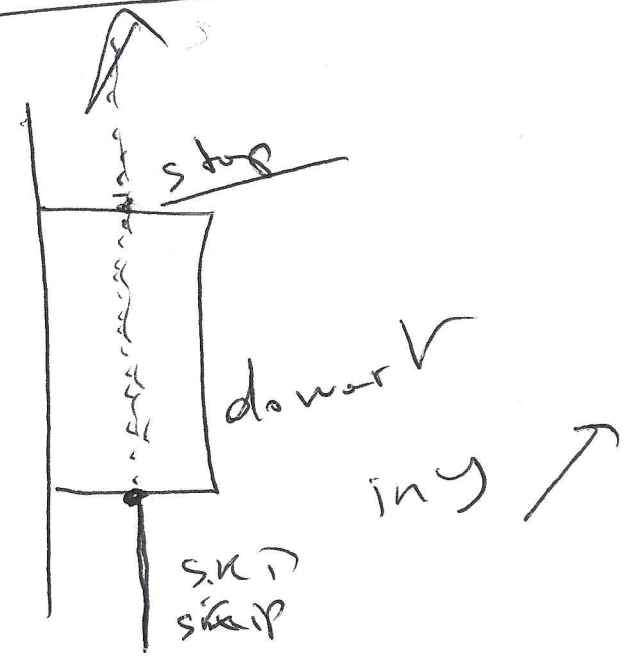
ordery in y.

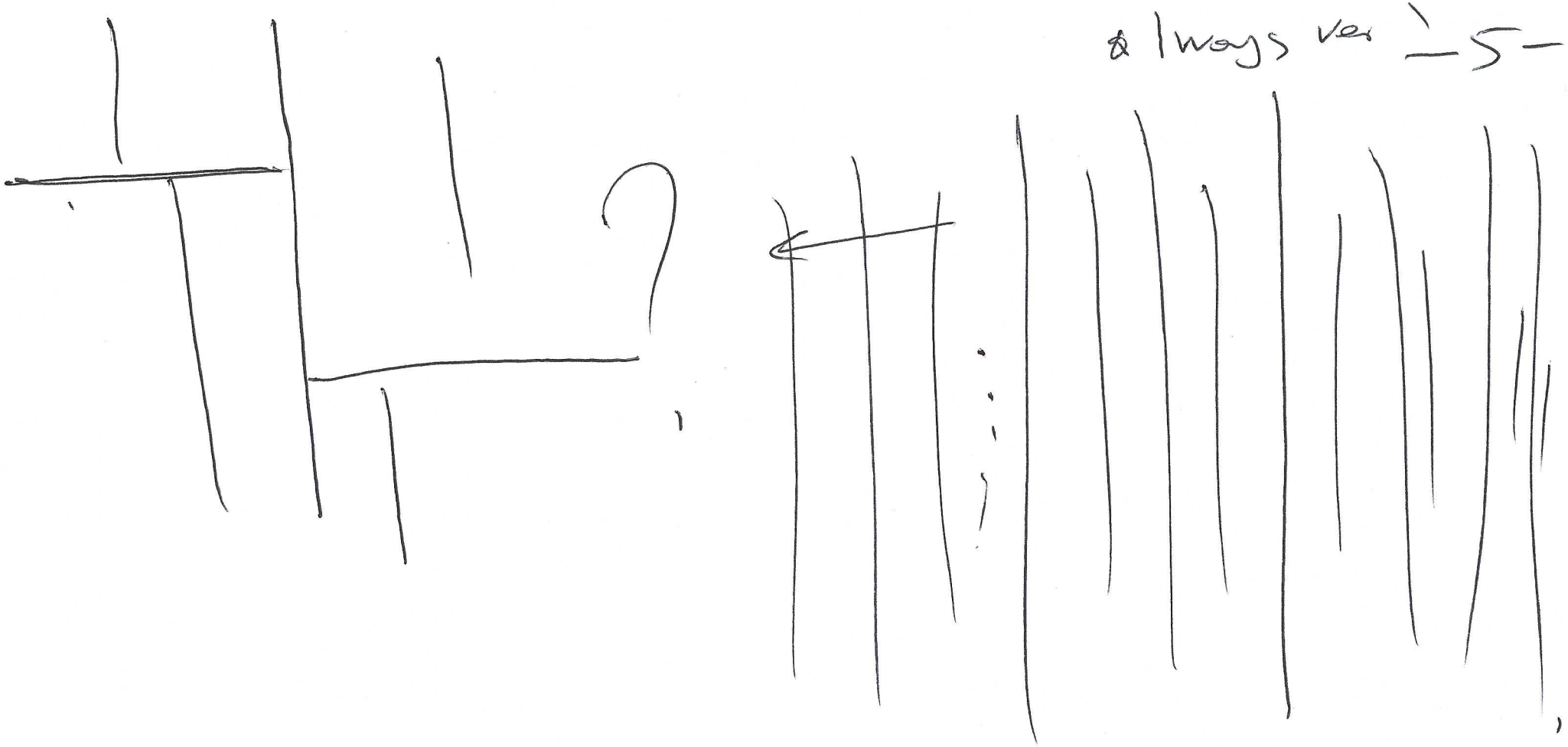
array of object $\{O_1\ O_2 \ldots O_n\}$

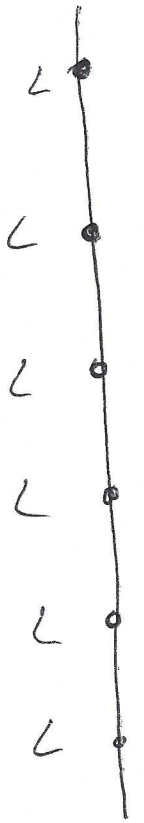
array of sorted ind. in x [9, 7, 2, 9, 1, 3 ...]

int

sorted in y [3, 9, 2, 0, 5 ...]







$x_{min} = x_{max}$

tie $p \in$ v.line

1) all go to Left (∞ rec) X

2) alternat < & R ✓

3) 1-dim closest pair.
sort then

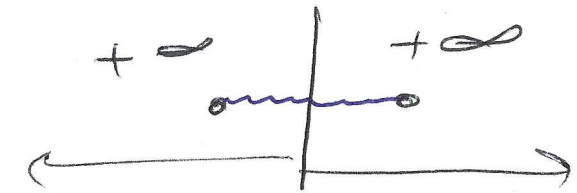
4) hor. split

~~base case~~

base case

~~n == 0~~

10% n == 1 | return $+\infty$

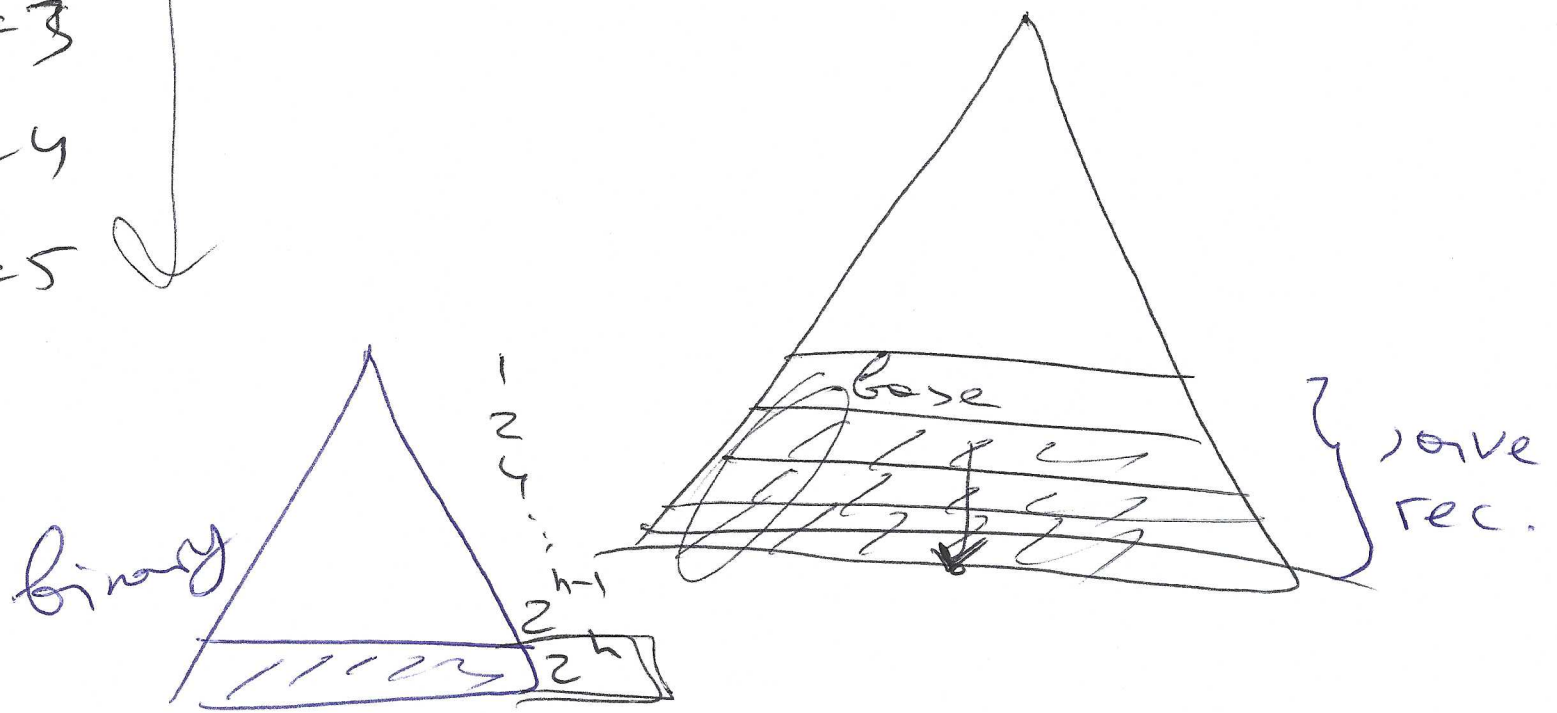


92% n == 2 | return $d(p_1, p_2)$

89% n == 3

n == 4

85% n == 5



$n = 2^{h+1} - 1$, half are leaves