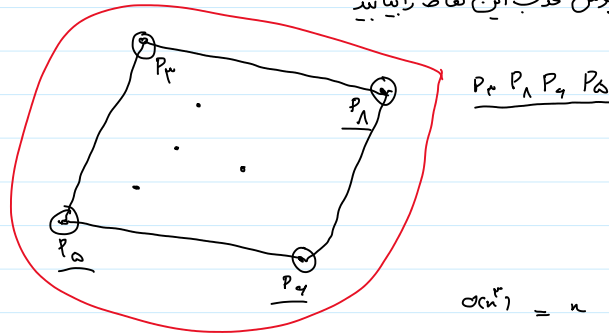


موضوع: تقسیم و حل - پوش محدب . الگوریتم هاروی

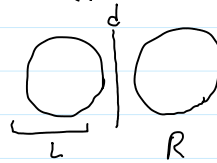
مساله [پوش محدب]: مجموعه $P = \{P_1, P_2, \dots, P_n\}$ شامل n نقطه داده شده، پوش محدب این نقاط را بیابید



ایده! نقاط را نصف کنیم برای $\frac{1}{2}$ قسمت حل کنیم و سپس جواب نهایی را ترکیب کنیم.

الگوریتم: $O(n \log n)$: همه نقاط را بر حسب x مرتب می کنیم. P_x

۱- خط عمودی d که نقاط را به $\frac{1}{2}$ زیر مجموعه با اندازه برابر تقسیم می کند می یابیم
۲- $O(n)$ به L_x و R_x را تشکیل دهد



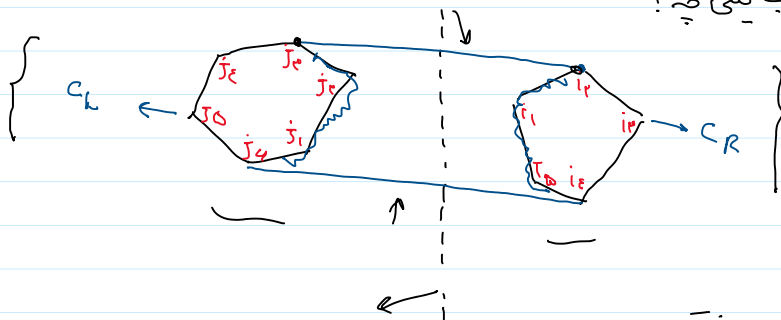
۳- به طور بازگشتی پوش محدب نقاط R و L را بیابید

۴- C_L و C_R را ترکیب و جواب نهایی را به دست می آوریم

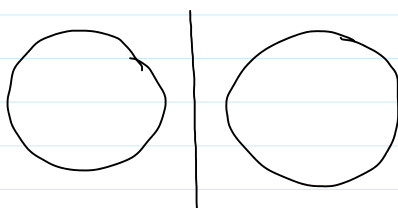
$$T(n) = 2T\left(\frac{n}{2}\right) + \boxed{}$$

زمان اجرا: $O(n \log n)$ ← مرحله ۴ در $O(n)$ انجام شود.

ترکیب دو پوش محدب یعنی چه؟

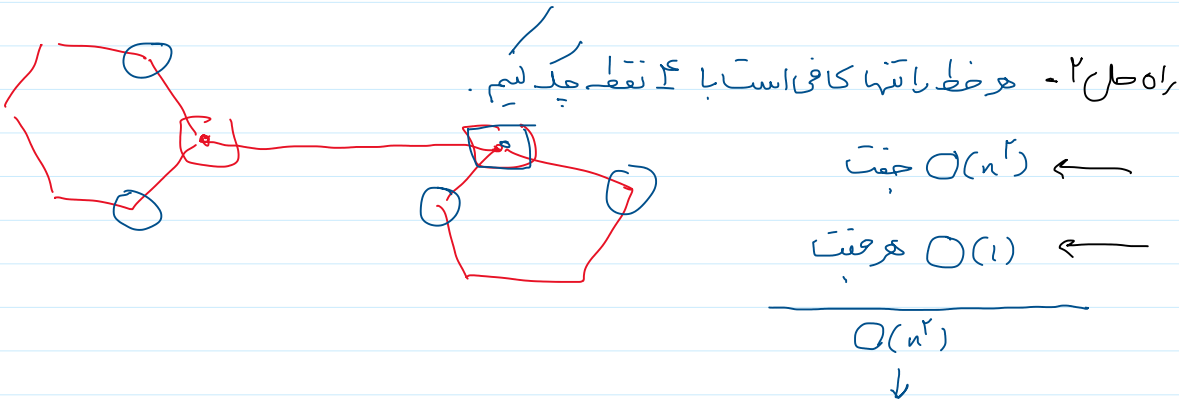


راه حل ۱) تمام رئوس های روی پوش محدب چپ و راست را چک می کنیم.



$$\left\{ \begin{array}{l} O(n^2) = \frac{n^2}{2} = \frac{n}{2} \times \frac{n}{2} \\ O(n) : \text{چک کردن هر نقطه} \leftarrow ? \\ O(n^3) \end{array} \right.$$

$$T(n) = 2T\left(\frac{n}{p}\right) + O(n^3) \rightarrow O(n^3)$$



$$T(n) = 2T\left(\frac{n}{p}\right) + O(n^2)$$

$$= T(n) = O(n^2)$$

راه حل ۳ -

← تعداد جفت ها $O(n)$

$O(1)$

a : rightmost point of C_L

b : leftmost point of C_R

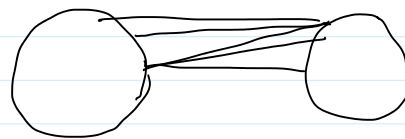
while \vec{ab} is not the upper tangent of C_L, C_R

while \vec{ab} passes through C_R

$b \leftarrow b.next \ // CCW$

while \vec{ab} passes through C_L

$a \leftarrow a.prv \ // CCW$



$$O(n) = O(1) \times O(n)$$

تعداد جفت ها \times چید کردن

$$T(n) = 2T\left(\frac{n}{p}\right) + O(n) = O(n \log n)$$

* ارجای الگوریتم متوقف شود جواب را یافته ایم : proof sketch

* از بین c_L و c_R یکی زودتر به راس دست می‌رسد.

← بابت تشخیص دهم از آن راس عبور نمی‌کند.