Homework 4: Voronoi Diagram in the Laguerre Geometry 閱讀心得

R10942152 游家權

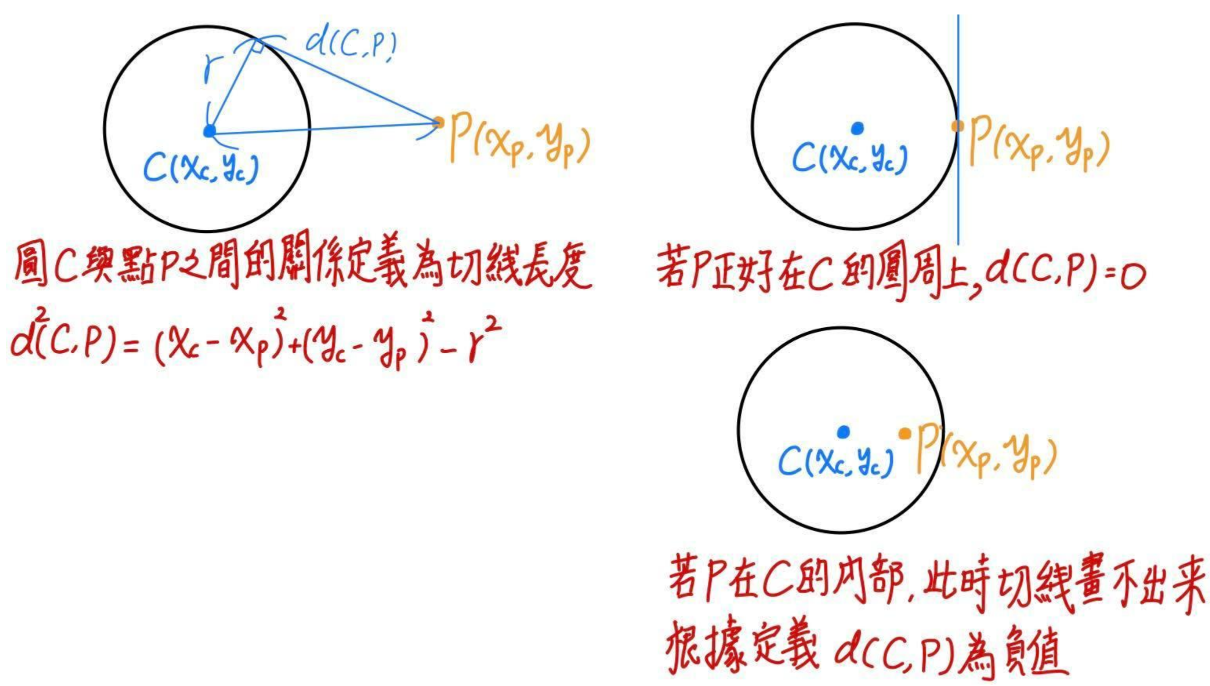
一、問題定義

本篇論文延續Shamos等人的Voronoi Diagram的研究，希望可以將二維Euclidean space的Voronoi演算法，延伸到Laguerre Geometry上面，並且維持時間複雜度仍是

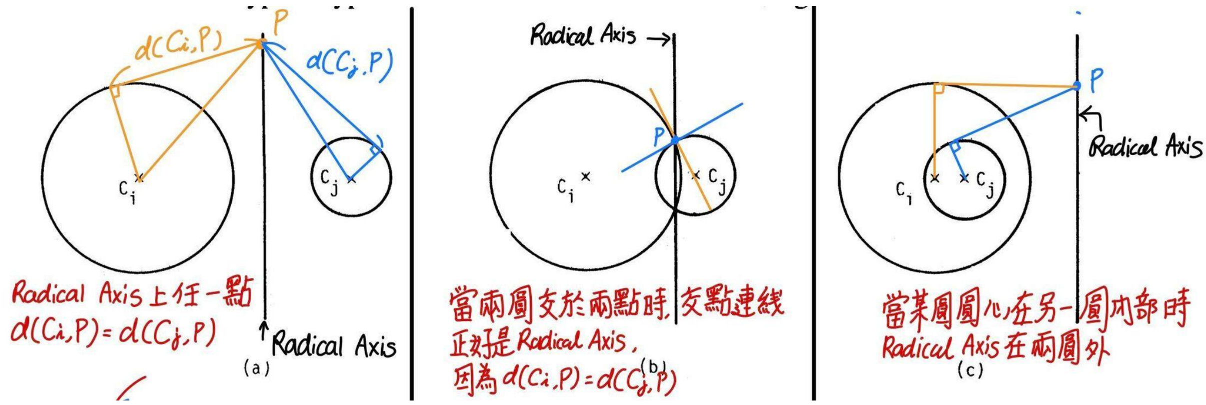
$O(nlogn)$。因此我們需要先定義何謂Laguerre Geometry。

Laguerre Geometry:

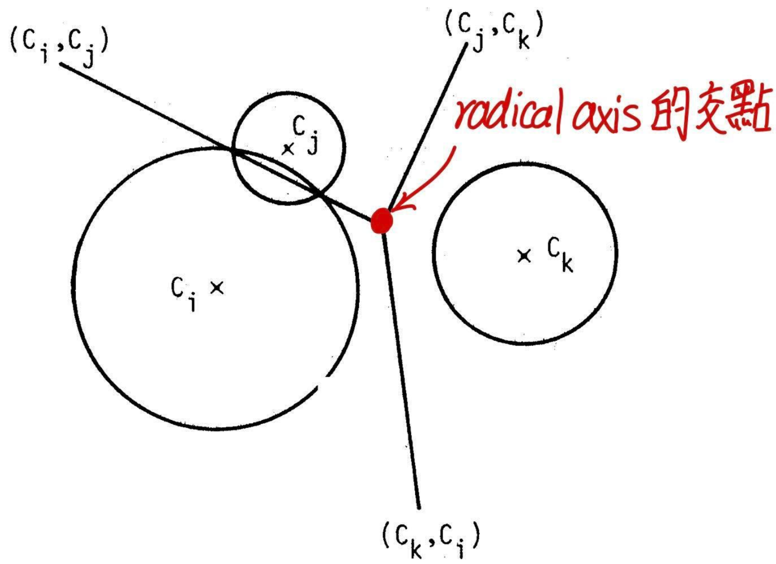
Laguerre Geometry中的一點C有三個變數(x, y, z)，這個點C對應到Euclidean space會變成一個有向圓，其中x, y代表圓心座標，z代表圓的半徑(但可以是負數)。而Euclidean space上的一個點P，到這個有向圓的距離也有特殊定義：此距離是點P到圓C的切線長度。



Laguerre Geometry中兩點間的中垂線，在Euclidean space中等同於找兩個有向圓的Radical Axis，Radical Axis線上任一點到兩個圓是等距的。範例如下圖。



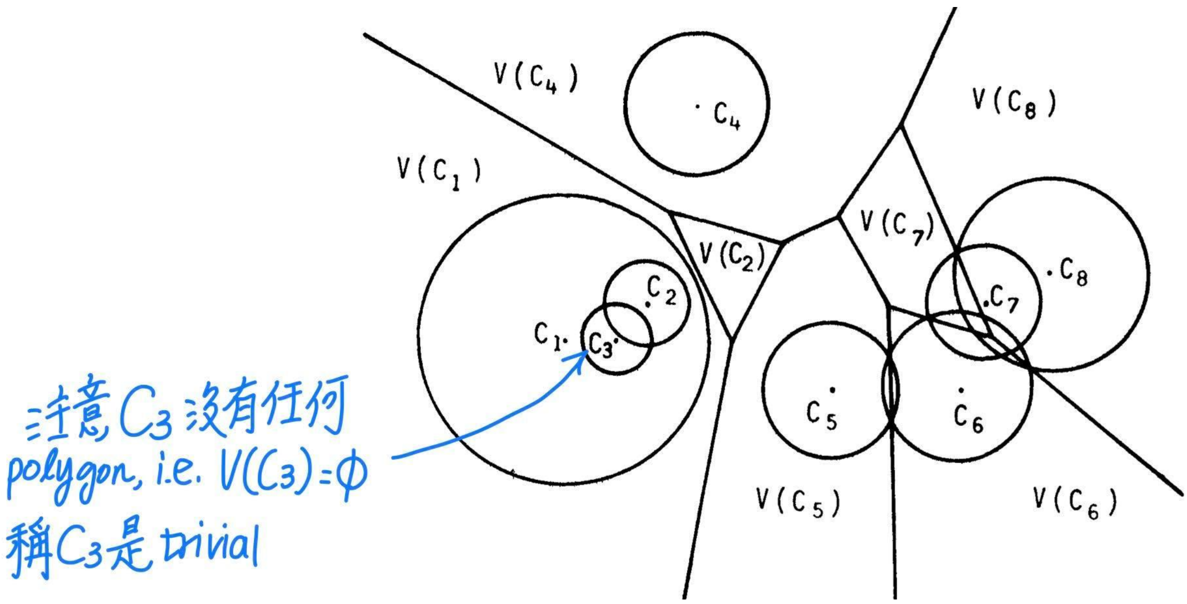
如同原本的Voronoi diagram上是由中垂線構成的圖形，我們定義在Laguerre Geometry中Voronoi diagram就是由這些Radical Axis所構成的圖形。本篇論文要解決的問題，就是想辦法找到Laguerre Geometry中的Voronoi diagram。下圖呈現由三個圓形成的Voronoi diagram的一個例子。



二、觀察問題

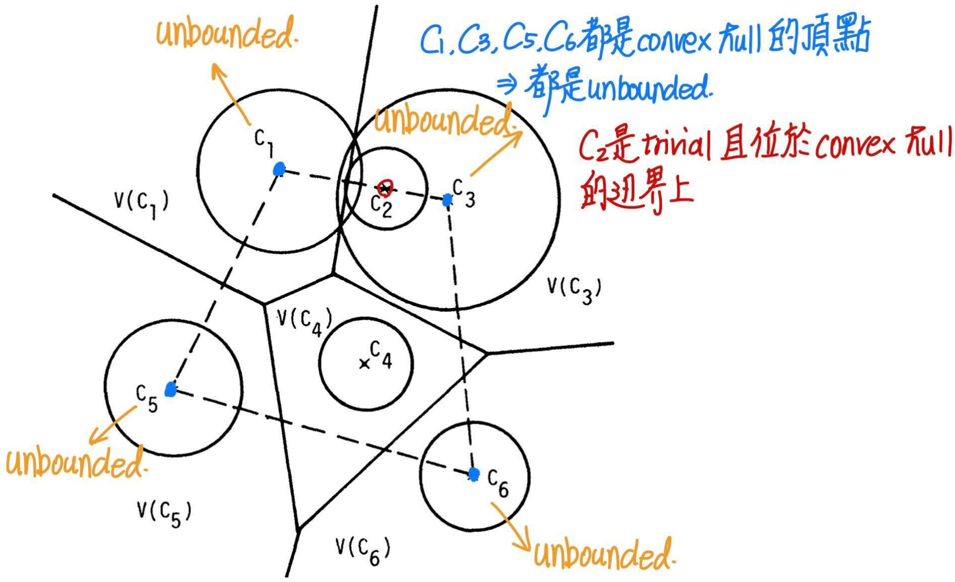
<u>Observation 1: Laguerre Geometry 的Voronoi diagram與Euclidean space的Voronoi的相同之處</u>

下圖是Laguerre Geometry 的Voronoi diagram的一個例子，可以注意到圓心可以不在它形成的Voronoi Polygon內部，甚至兩者可以沒有任何交集(e.gC_2 and $V(C_2)$)，這種沒有交集的C_i我們稱它為Improper。而完全無法形成任何Polygon的圓稱為Trivial。

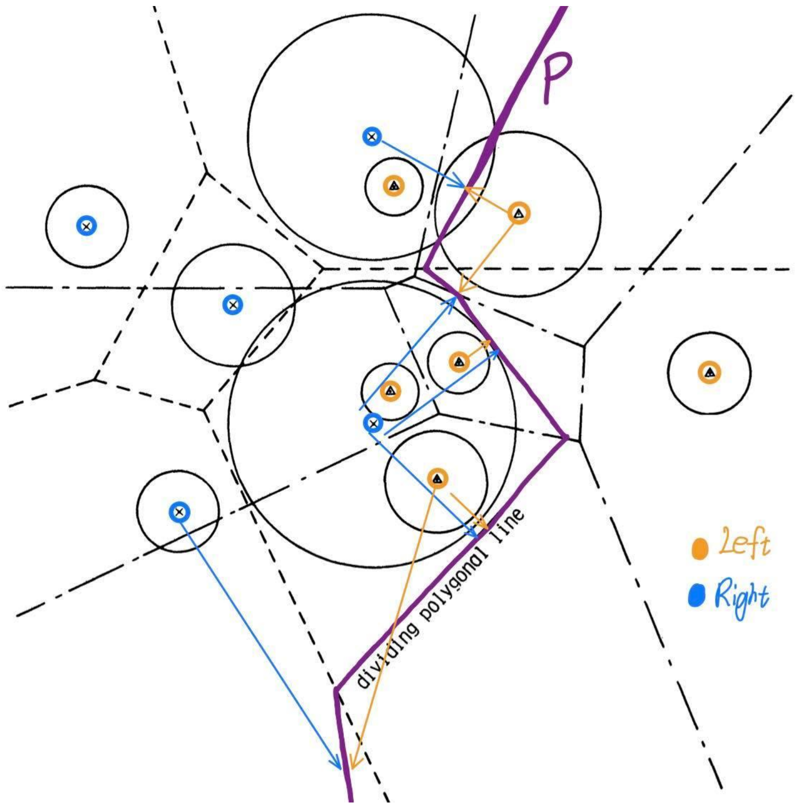


Laguerre Geometry的Voronoi diagram 跟原版有很多相同之處，包含Voronoi diagram edge與point的個數是$O(n)$，這個性質讓我們可以直接套用原本的Voronoi diagram演算法而不需要付出額外的時間成本。

另外，點所形成的Convex Hull上的頂點，它們的Polygon都會往無限遠處延伸，也就是他們的polygon unbounded，跟原版的演算法一樣，我們也可以利用這個性質來找dividing line的兩條射線。下圖是的Voronoi diagram 中Convex Hull的例子，需要注意C_2是位於Convex Hull上的邊界，同時也是Trivial的圓。

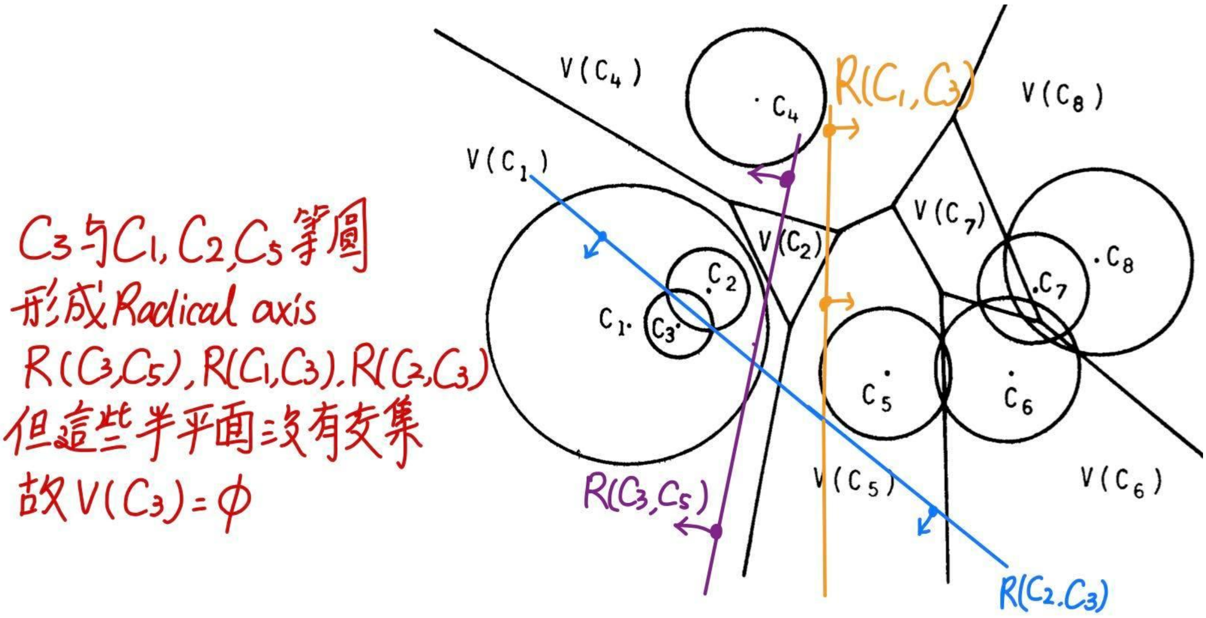


最後一個相同之處，也是最重要的。就是在做Merge的時候Laguerre Geometry的Voronoi diagram基本上做法也是跟原版一樣。一樣可以去找一條單調往下的Dividing Line(下圖紫色線)，這條Dividing Line會是由左右兩個小的diagram內的兩點形成的Radical Axis組成的，其中包含上下兩條射向無窮遠處的射線。只是圖畫出來之後會發現Dividing Line並不會像原版一樣一定能將左側的點分到左邊，右側的點分到右邊，而是左右都有可能。雖然有點違反直覺，不過根據Radical Axis的定義，當某圓位於另一圓的範圍內時，就會發生這種情形，下圖將Dividing Line的每個線段是哪兩個圓的Radical Axis都標示出來，便於理解。



<u>**Observation 2: Laguerre Geometry 的Voronoi diagram與Euclidean space的Voronoi的相異之處**</u>

Laguerre Geometry中絕大部分的特性都跟原本相同，但是有一個性質特別麻煩，就是Laguerre Geometry 的Voronoi diagram可能出現有些圓C 是Trivial，也就是完全沒有任何對應的Polygon，例如下圖中的C_3就是如此，因為C_3離C_1，C_2太近導致影響力被完全吃掉，無法生成出任何的Polygon，下圖畫出了C_3與各個鄰近圓的Radical Axis，說明了C_3沒有Polygon 的原因。

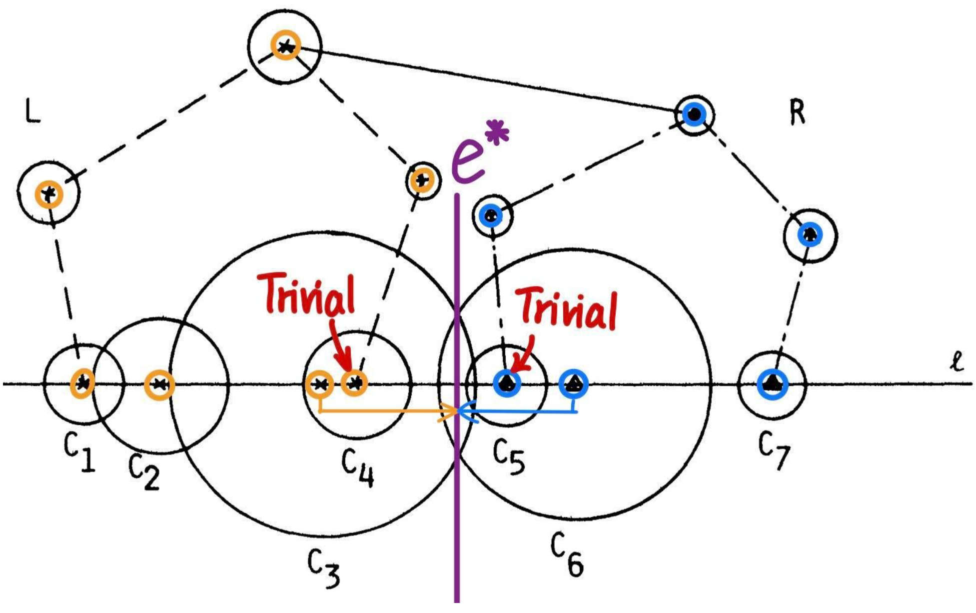


這個Trivial圓的性質非常棘手，是原版的Vonoroi Diagram演算法沒有考慮到的情形。原版的Vonoroi Diagram是不管點之間離的多近，都一定會有一個區域是專屬於那個點，沒有了這個性質，對於Vonoroi Diagram找射線的步驟會形成漏洞，而本篇論文就是找到了一個方法可以在線性時間內處理掉這個問題。

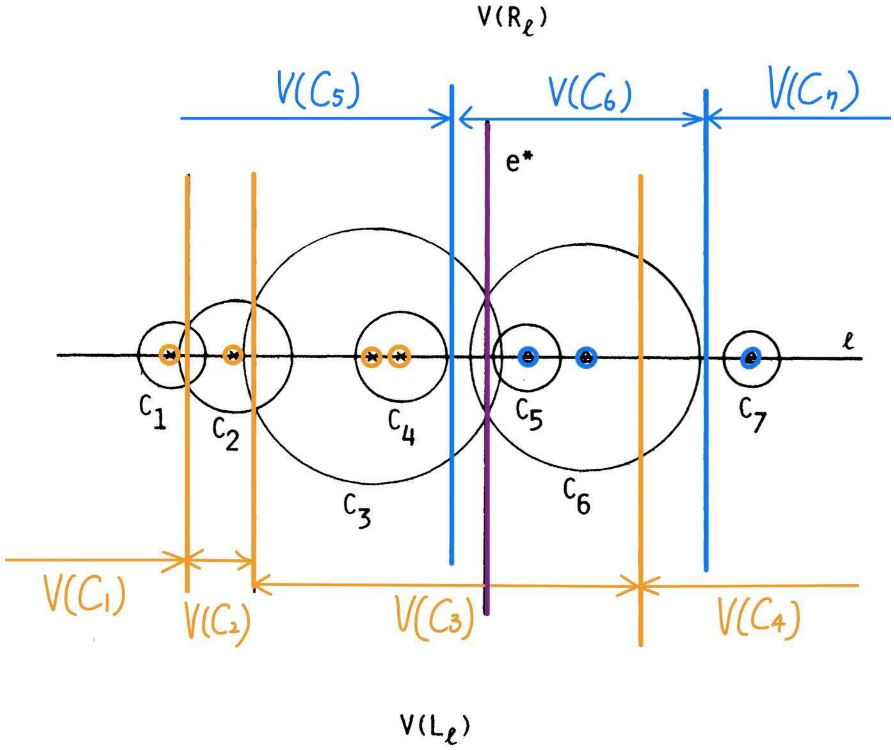
<u>Observation 3: 在線性時間內找到Vonoroi Diagram的射線</u>

在原版的演算法裡，當我們想要找Dividing line的射線的時候，我們是直接去找左半邊點的Convex Hull($CH(L)$) 跟右半邊點的Convex Hulll($CH(R)$)，然後把左右兩個Convex Hull Merge成一個大的Convex Hull，並且找到需要新增的線段的兩端點，而這兩個端點就是所形成的中垂線，就是我們要的射線。

上述演算法在原本的問題可以解得很好，但是在Laguerre Geometry裡就不成立。如下圖所示，橘色的點是左半邊的點，藍色的點是右半邊的點，而線l就是merge成大的Convex Hull 需要補的線段，而很不幸的這條線的兩端點是C_4, C_5，兩個圓都是Trivial，也就是無法形成任何Polygon，所以當然也無法形成任何射線。而在這個例子裡，真正該選的端點是C_3, C_6，這兩個圓形成的Radical Axis e^*才是屬於Dividing Line。



改進的演算法是把線l上的圓跟各別左右半邊的Voronoi Diagram畫出來，如下圖所示，橘線代表左邊的Voronoi Diagram，藍色代表右邊的。此時沿著線段l，從左到右搜尋一次，尋找有沒有剛好哪個位置會使得左右兩邊距離相同，找到這個點等同於找到Radical Axis，也就是找到射線。由於線段l上最多n個點，且這個方法也只會搜尋一次，所以這個找射線的方法依然是花費$O(n)$的時間。



三、解法敘述

Step1 Divide: 找X座標的中位數，並且用這個中位數將所有點分成左右兩邊，分別去解左右兩個小集合的Voronoi diagram。

Step2 Conquer: 一直拆分問題直到**problem size**小到可以被直接解出來。

Step3 Merge: 把左右兩個小問題的Voronoi diagram合起來，這時需要尋找兩個diagram之間的Dividing Line。

Step4 Find Ray: 找上下兩條射線剛好會是Dividing Line的起點與終點，這裡需要使用**Observation 3**的方法。

Step4 Construct Dividing Line: 沿著起始的射線前進，如果遇到邊界，就換掉其中一個形成**Radical Axis**的點，這樣**Radical Axis**會形成轉折，重複上述步驟直到遇到Dividing Line的終點射線。

Step5 End: 把得到的Dividing Line放進Merge後的Voronoi diagram，並切掉多餘的邊，就可以回傳函數，解出Problem size最大的答案。

時間複雜度分析:由於在所有解題的步驟中, Laguerre Geometry的時間複雜度跟原版的完全一致, 所以理所當然時間複雜度也是相同:$O(nlogn)$

由於一個Voronoi diagram會被拆成兩個小問題, 而把兩個小問題Merge回來需要花$O(n)$,因此可得遞迴式:$T(n) = 2(T/2) + O(n)$, 解這個遞迴式就可以得到$O(nlogn)$

四、讀後心得

Voronoi diagram乍看之下會覺得是非常困難的問題, 不僅不容易理解也不容易畫圖。但是這麼難的問題竟然可以用簡單的D&C解, 實在是有點神奇。

這篇論文一開始提到的Laguerre Geometry, 因為之前完全沒聽過, 所以花了很多時間理解, 不過讀到一半大概開始理解到這個作者可能是想要做跟圓有關的問題, 而這個Laguerre Geometry對他來說是一個把圓對應成點的好用的工具。對我來說最難理解的兩個圓互相包含的時候, Radical Axis會落在兩圓之外的情形, 不僅違反直覺也很難作圖。

但這篇論文也向我展示了如果選對工具, 要拓展已知的算法是一件簡單的事。從Euclidean space變成Laguerre Geometry, 乍看之下改變了很多, 但是經過論文裡一個一個Lemma的推導, 很快就發現其實兩者的性質是非常接近的, 演算法上幾乎不用修改任何東西, 只有最後找射線的這個部分因為出現trivial 圓的關係需要進行小修改而已, 十分令人驚奇。