

演算法作業一

想法:

因為最初暴力解法是遍歷所有點，所以複雜度是 $O(n^2)$

但是我們想用更有效率的解法例如把問題縮小一半再去比較最短距離的點對

實際作法步驟:

把所有點先用 x, y 座標由小到大排序(這邊採用 x 比較直觀)

找出中間點 m 再將

左半邊的點對比較排序找出最短的距離 d_l

右半邊的點對比較排序找出最短的距離 d_r

$$\delta = \min\{d_l, d_r\}$$

Interval: $(m - \delta, m + \delta)$

由於當我們這樣畫定區間之後，可以把剛剛未考慮的橫跨

兩區域的點對搜尋，大於 δ 範圍的不需考慮，由於我們只

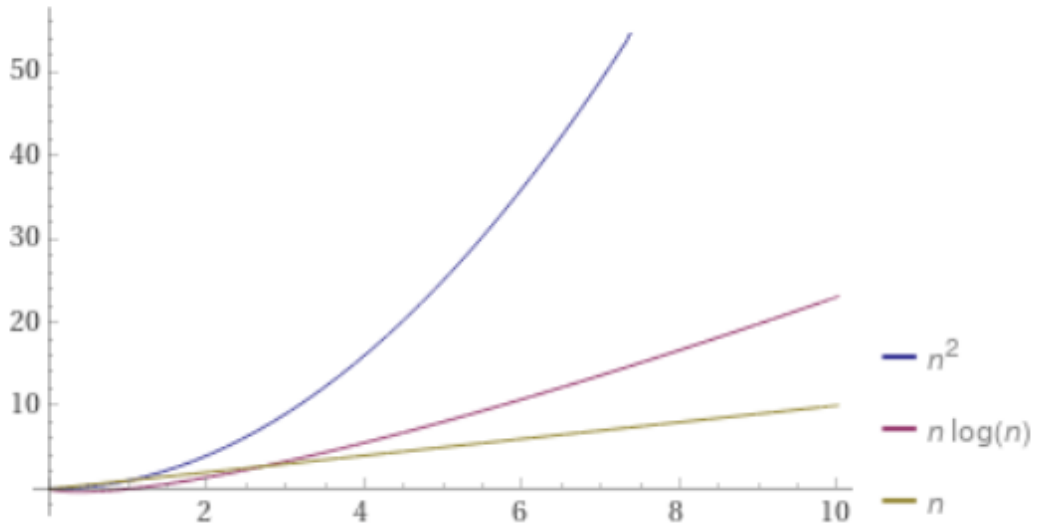
取最小值所以將中間點 $middle$ 為中心， δ 當半徑來界定範圍，將所有其中的點對搜尋出最小值再來跟 d_l, d_r 做比較

得出最短路徑

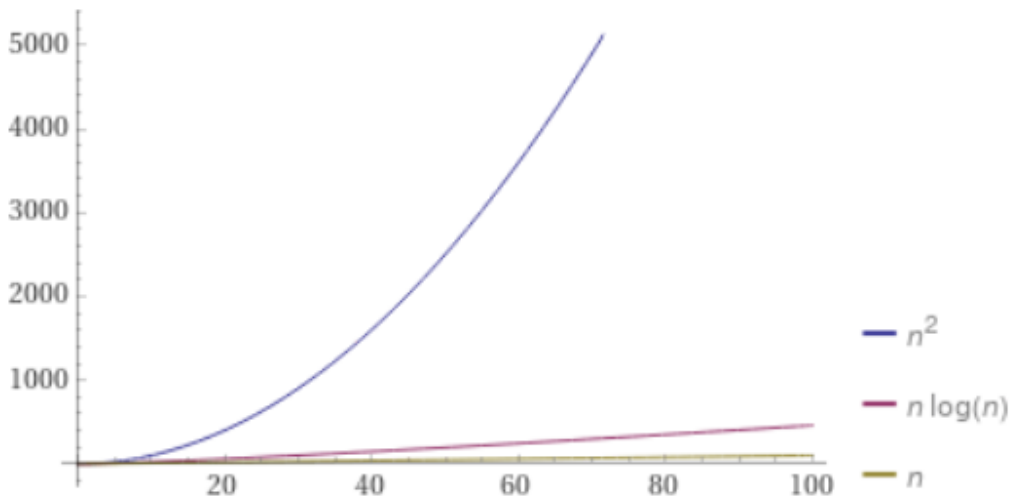
複雜度減為: $O(n \lg n)$

閾值(threshold)決定值

Plot



Plot



當 $n=2$ 時由於直接遍歷的結果只有一種，跟我們用 **divide and conquer** 的過程一樣甚至還有點多餘，因為就只有兩點直接計算兩點距離就是最短路徑了，所以這裡建議用直接搜尋方法

Threshold 取值原因在於當我們將所有點對排序之後

已經一分为二，左右各是一部分

當 $n=3$:

雖然 $O(n^2) > O(n \lg n)$ ，理論上使用 divide and conquer 會比較省時，但是實際原本單純用 3 次的搜尋結果排序找出最小的跟我們用上述思路解題:找出 3 個之中的中位數，最短距離一定是中位數與剩下兩數距離的最小者，所以還是比較有效率一點，儘管沒有明顯的速度提升

當 $n=4$

雖然 $O(n^2) > O(n \lg n)$ ，理論上使用 divide and conquer 會比較省時，但是實際原本單純用 6 次的搜尋結果排序找出最小的跟我們用上述思路解題:找出 4 個之中的中位數，1 跟 23 跟 4 分別找距離，最後看 2 跟 3 是否落在中間帶狀範圍內如果有則多一次，所以還是比較有效率一點

自此，由上得知當 $n \geq 3$ 時候， $O(n^2)$ ， $O(n \lg n)$ 之間的成長速率已經漸漸拉遠，意味著兩者複雜度的搜尋方法數及時間上來說， $O(n^2)$ 所花時間會越來越多，根據上面提供的圖形所示， $O(n^2)$ 指數型成長已經漸漸拉大計算時間，相比於 $O(n \lg n)$ 雖然不是線性成長但也是逼近線型函數 n 的成長圖形，所以我們若是 $n \geq 3$ 來看，確實用 divide and conquer(上

面第一部分提到的演算法)會相對快速許多，節省時間

實驗數據:

(n=2)

暴力解法遍歷所有點對的 $O(n^2)$ 所花費時間較少

2.268s vs 2.931s 但是實際上巨觀來看的話這個差距不大

```
C:\C\演算法作業1.exe
2
1 2
3 4
暴力解,  $O(n^2)$ 
2.828
Process returned 0 (0x0)   execution time : 2.268 s
Press any key to continue.
```

```
C:\C++\Projects\演算法作業1.exe
2
1 2
3 4
分治法  $O(n \lg n)$ 
2.828
Process returned 0 (0x0)   execution time : 2.931 s
Press any key to continue.
```

(n=4)

點對稱性(所花費時間差不多)

隨機不規律生成點對:暴力解時間依然需 10s+, 分治法卻仍可以減至 2.5s 內

```
C:\C\演算法作業1.exe
4
1 2
-1 -2
3 2
-3 -2
暴力解,  $O(n^2)$ 
2.000
Process returned 0 (0x0)   execution time : 11.189 s
Press any key to continue.
```

```
C:\C++\Projects\演算法作業1.exe
4
1 2
-1 -2
3 2
-3 -2
分治法  $O(n \lg n)$ 
2.000
Process returned 0 (0x0)   execution time : 12.729 s
Press any key to continue.
```

```
C:\C\演算法作業1.exe
4
-7 5
-3 11
0 8
1 6
暴力解,  $O(n^2)$ 
2.236
Process returned 0 (0x0)   execution time : 12.377 s
Press any key to continue.
```

```
C:\C++\Projects\演算法作業1.exe
4
-7 5
-3 11
0 8
1 6
分治法  $O(n \lg n)$ 
2.236
Process returned 0 (0x0)   execution time : 2.472 s
Press any key to continue.
```

(n=10)

隨機不規律生成點對:暴力解時間已經隨著點對數增加至 **40s+**，分治法卻仍可顯著減少至 **2s** 內完成
足以看出差異

```
C:\演算法作業1.exe
10
-7 4
-6 8
7 5
8 4
1 3
0 9
-6 2
-5 6
2 2
3 0
暴力解,0(n^2)
1.414
Process returned 0 (0x0)   execution time : 40.152 s
Press any key to continue.
```

```
C:\C++\Projects\演算法作業1.exe
10
-7 4
-6 8
7 5
8 4
1 3
0 9
-6 2
-5 6
2 2
3 0
分治法0(nlgn)
1.414
Process returned 0 (0x0)   execution time : 1.875 s
Press any key to continue.
```