資料探勘 Project3 報告

資訊三 E44051259 陳冠霖

**報告大綱：**

連結分析是一種用來評斷兩個nodes關係的技術，在本次的project中，我將實作三個連結分析的演算法亦即為PageRank、HITS、SimRank，前兩個演算法是搜尋引擎中用來分析網頁排序著名的演算法，後者則為分析節點相似度的演算法。

**實作詳細：**

一：Hits

Hits裡面包含兩個數值，一個是hub，一個是authority，而這兩個值，可以隨機給，一開始我是都給1，但這樣在迭代收斂的時候，似乎會怪怪的，所以最後我是給定亂數。而後每次迭代，hub的值為所有child的authority的值的合，而child的authority則為原本的parent的hub值，下去迭代值到收斂。我是設定early\_stopping = 10000，或則收斂的RMSE小於10^-15就停止。

二：pageRank

pageRank裡面只有一個值，一樣一開始我們也是給予隨機值，因為最後會收斂，所以沒有關係。此外他跟前面的hits相比多了一個參數，那就是damping factor。然後一開始一樣是先將我們的點讀入，並建立parent和child。變且把這全部的node變成一張圖，然後在依次迭代計算直到收斂為止。

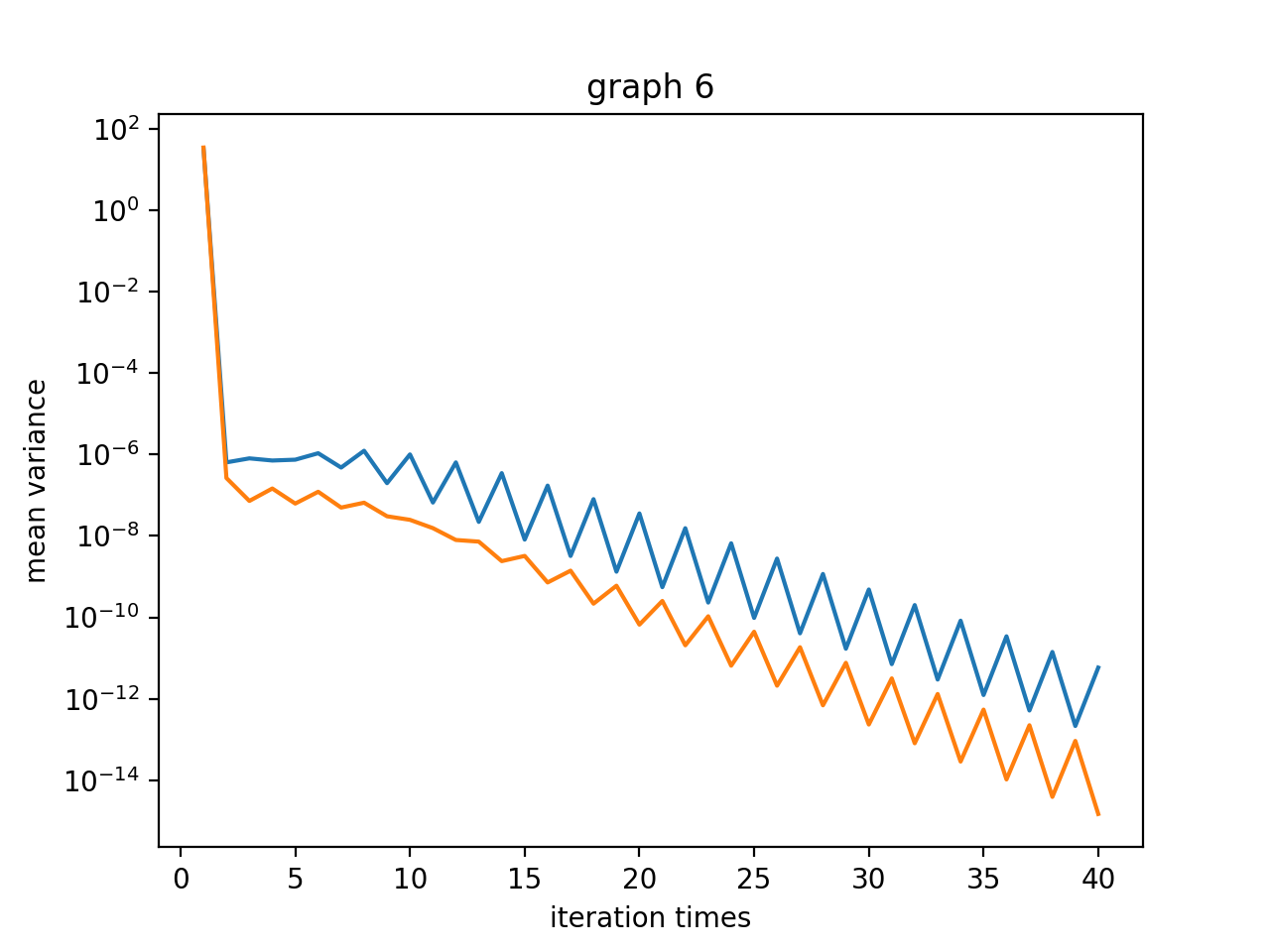
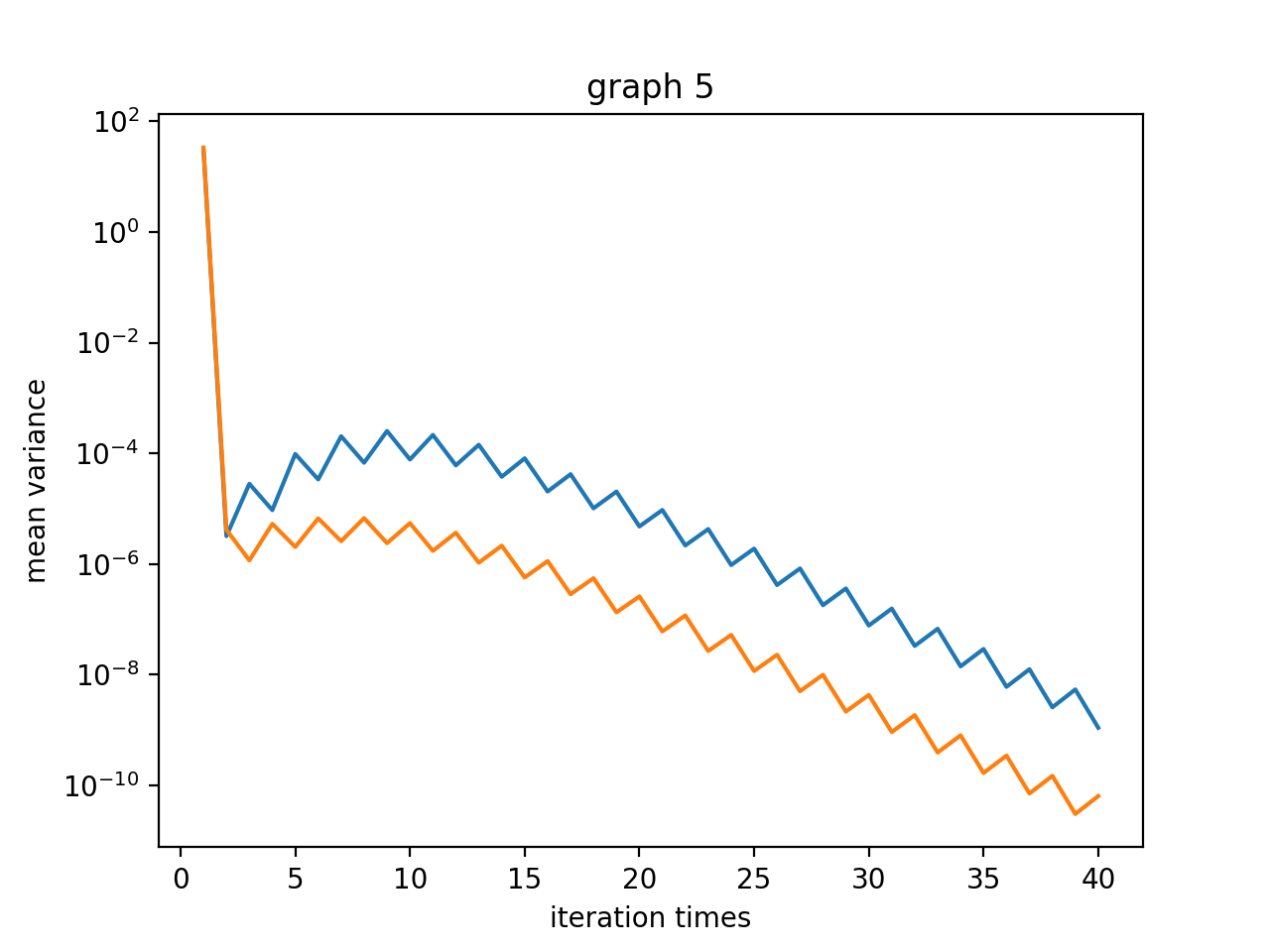
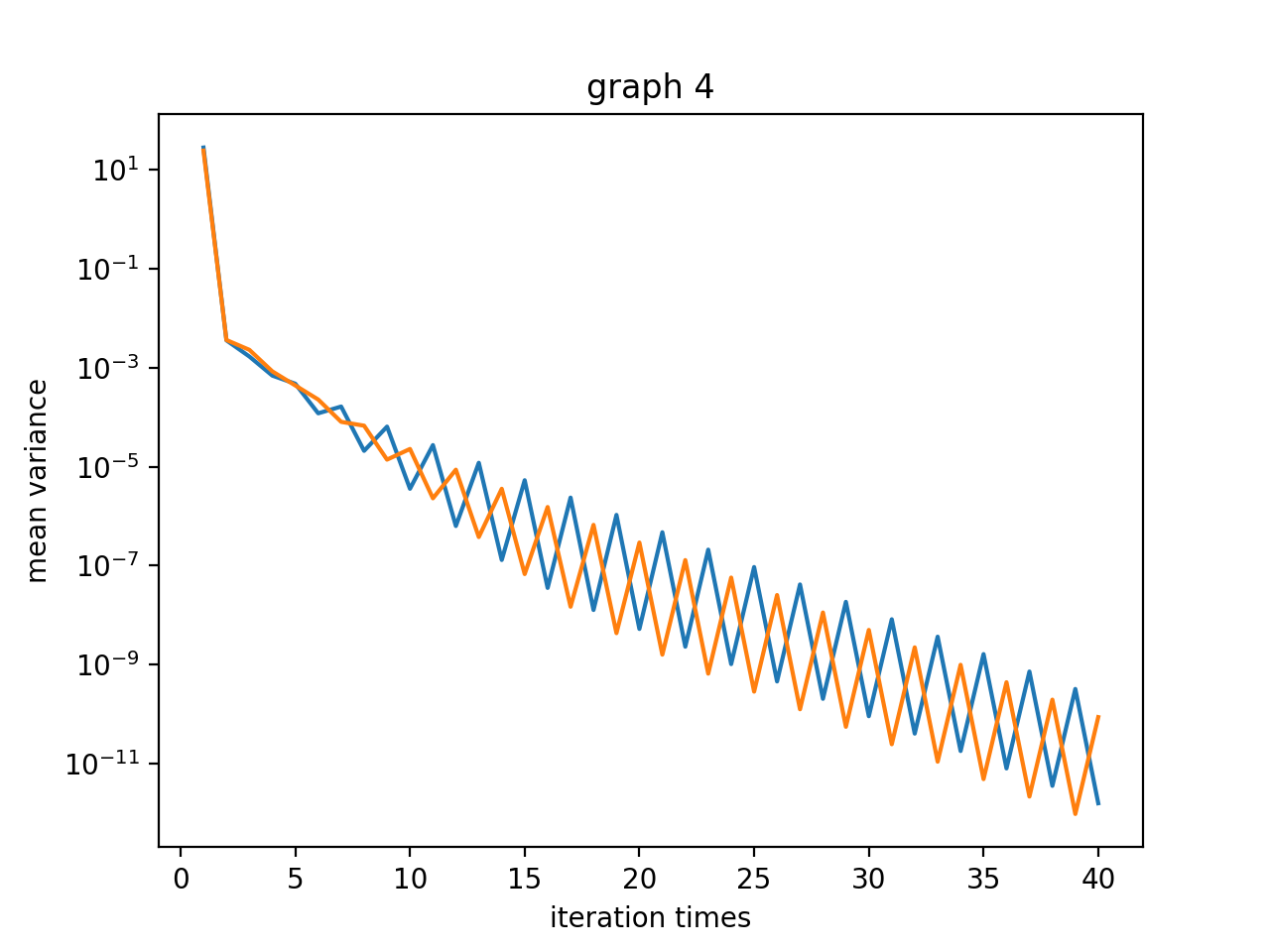
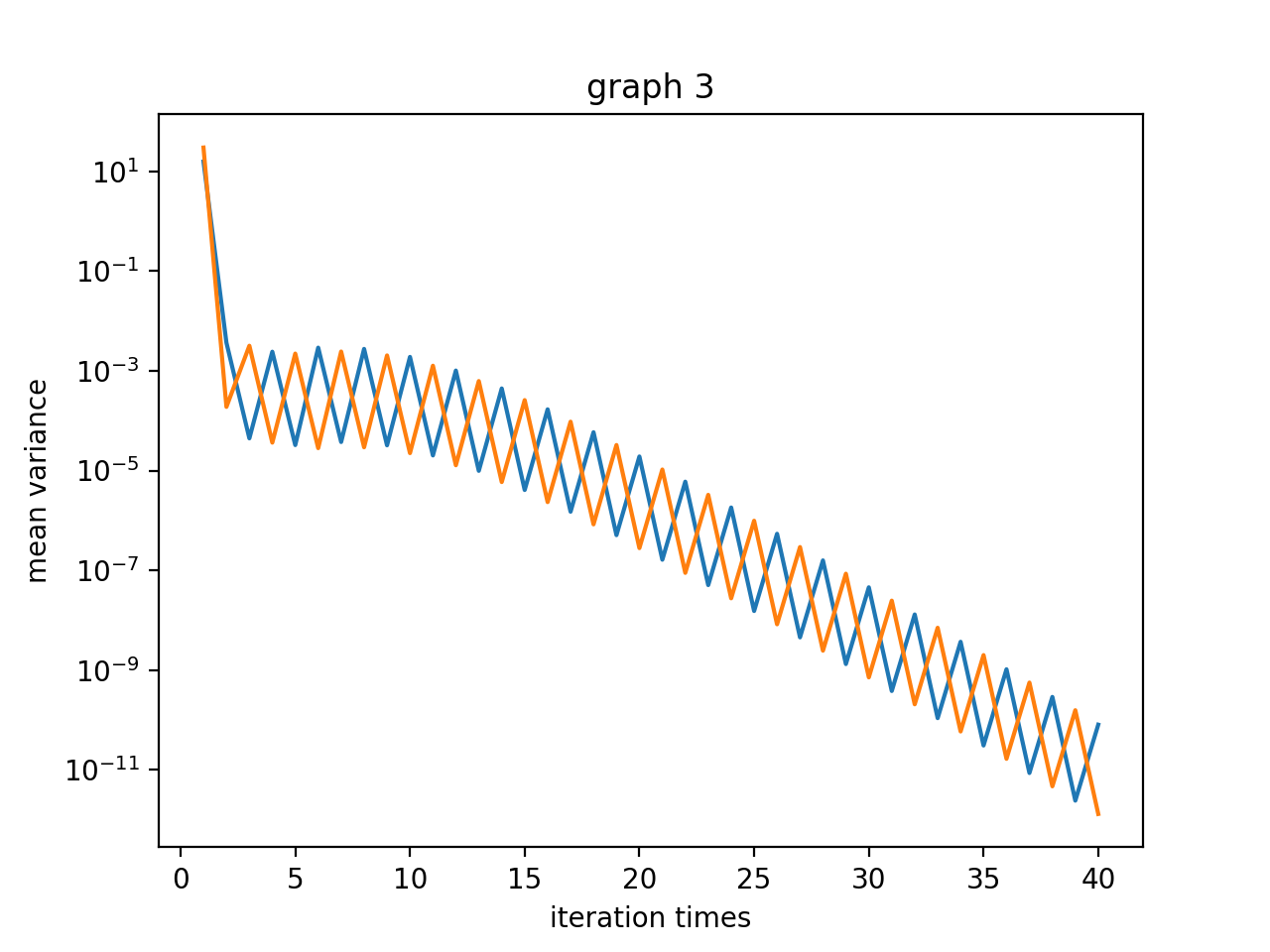
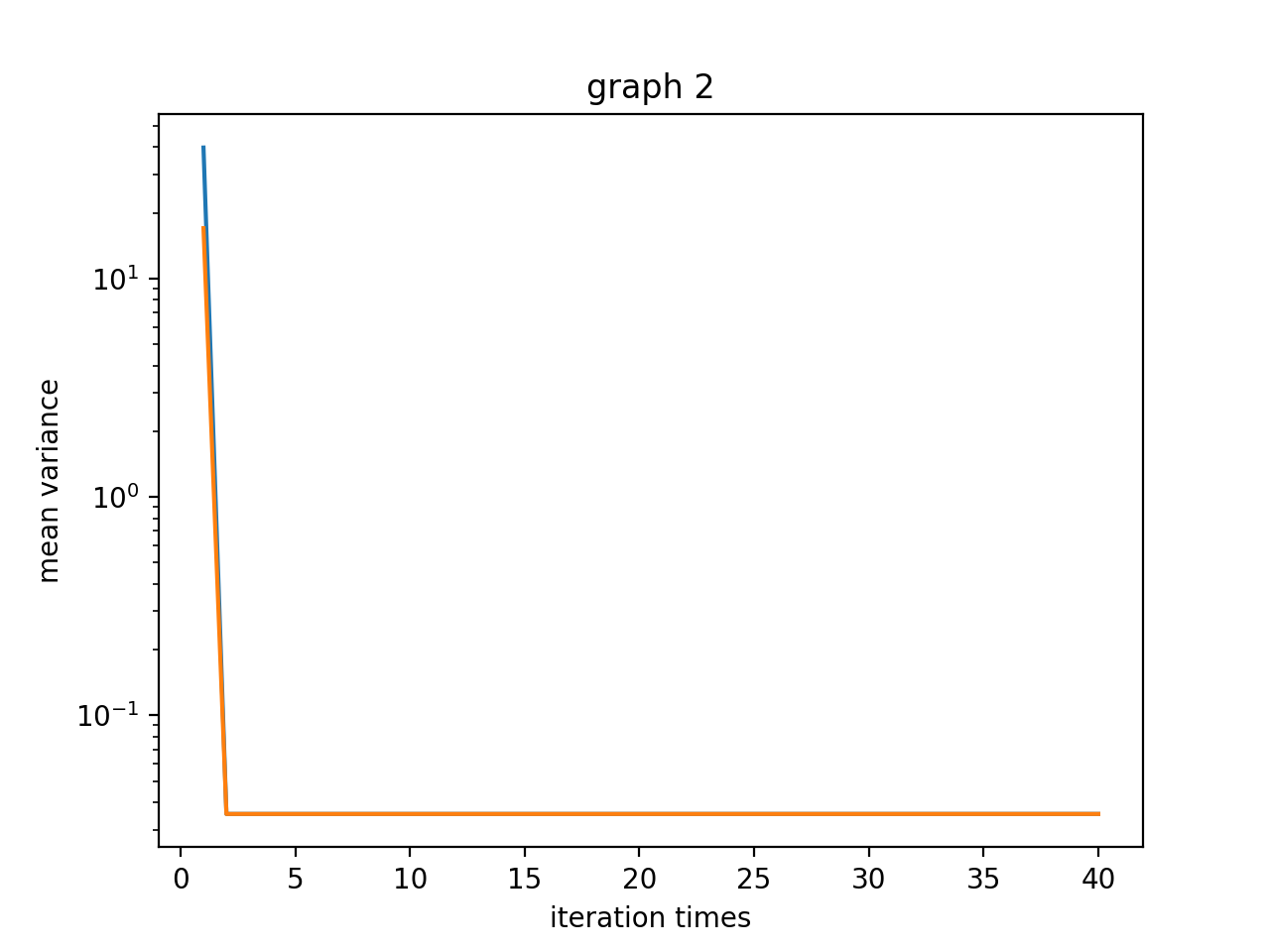
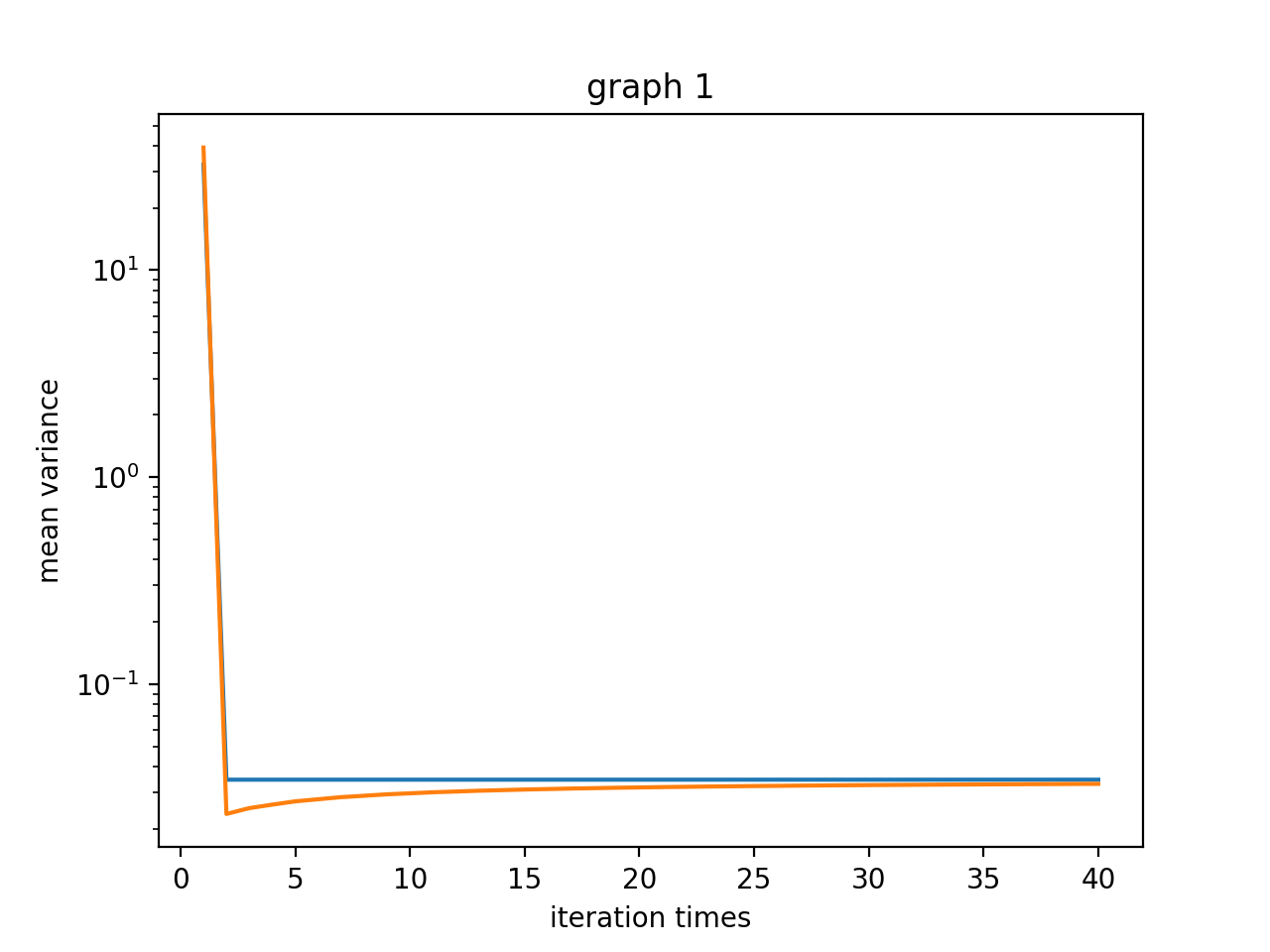
三：SimRank

用來計算兩個點的相似度分析，c值可以任意給定，我們是給定0.5，接著對於兩個點的分別的parent數量，然後在去計算他們的父母的點中，全部走訪一遍去計算他們的相似度，直到沒有父母（則相識度為0），而兩個同樣的點的相似度為1，之後再全部相加，接著除上這兩個點的parent數相乘，回傳就是答案了。

**分析與討論：**

一：Hits

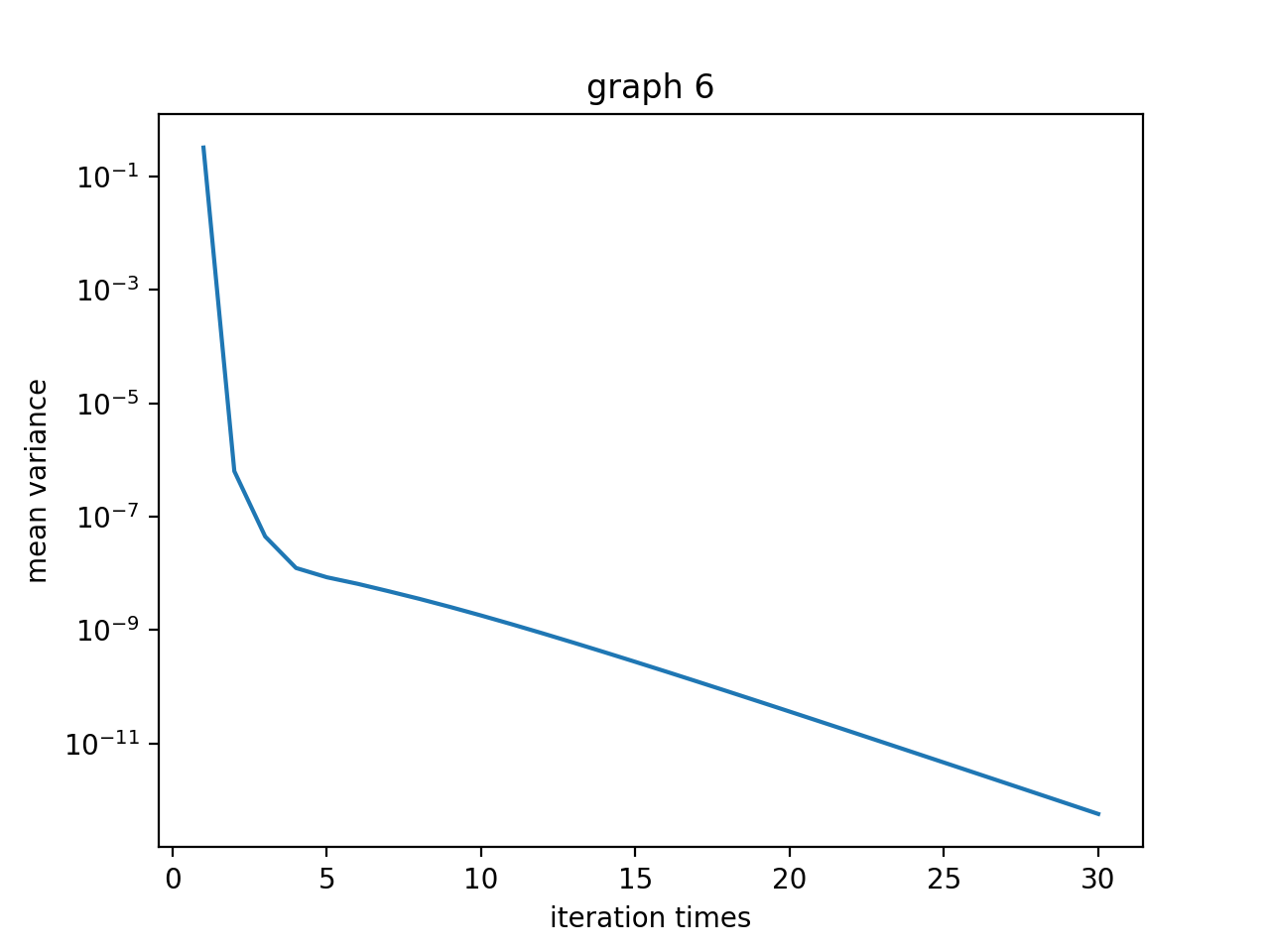
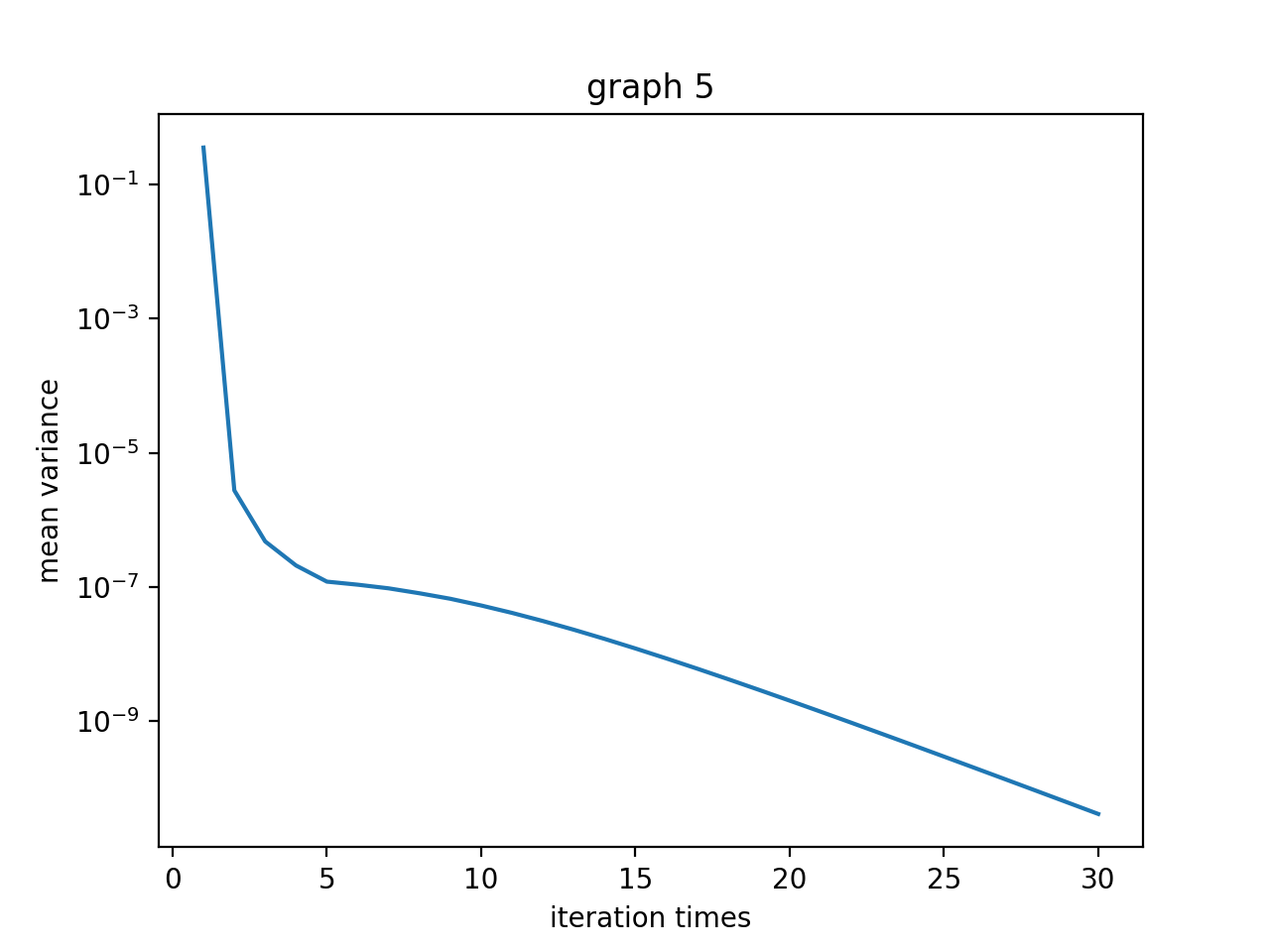
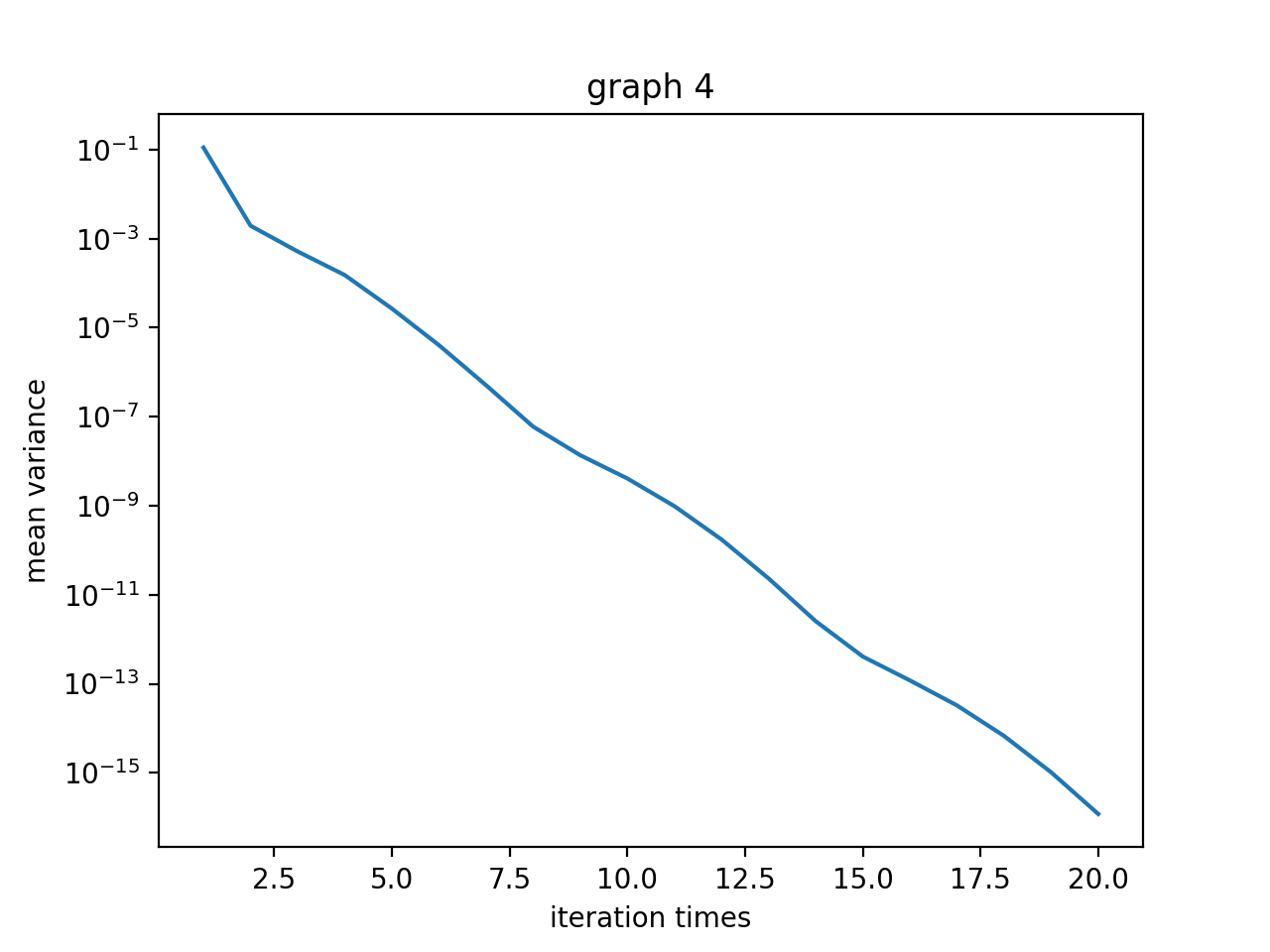
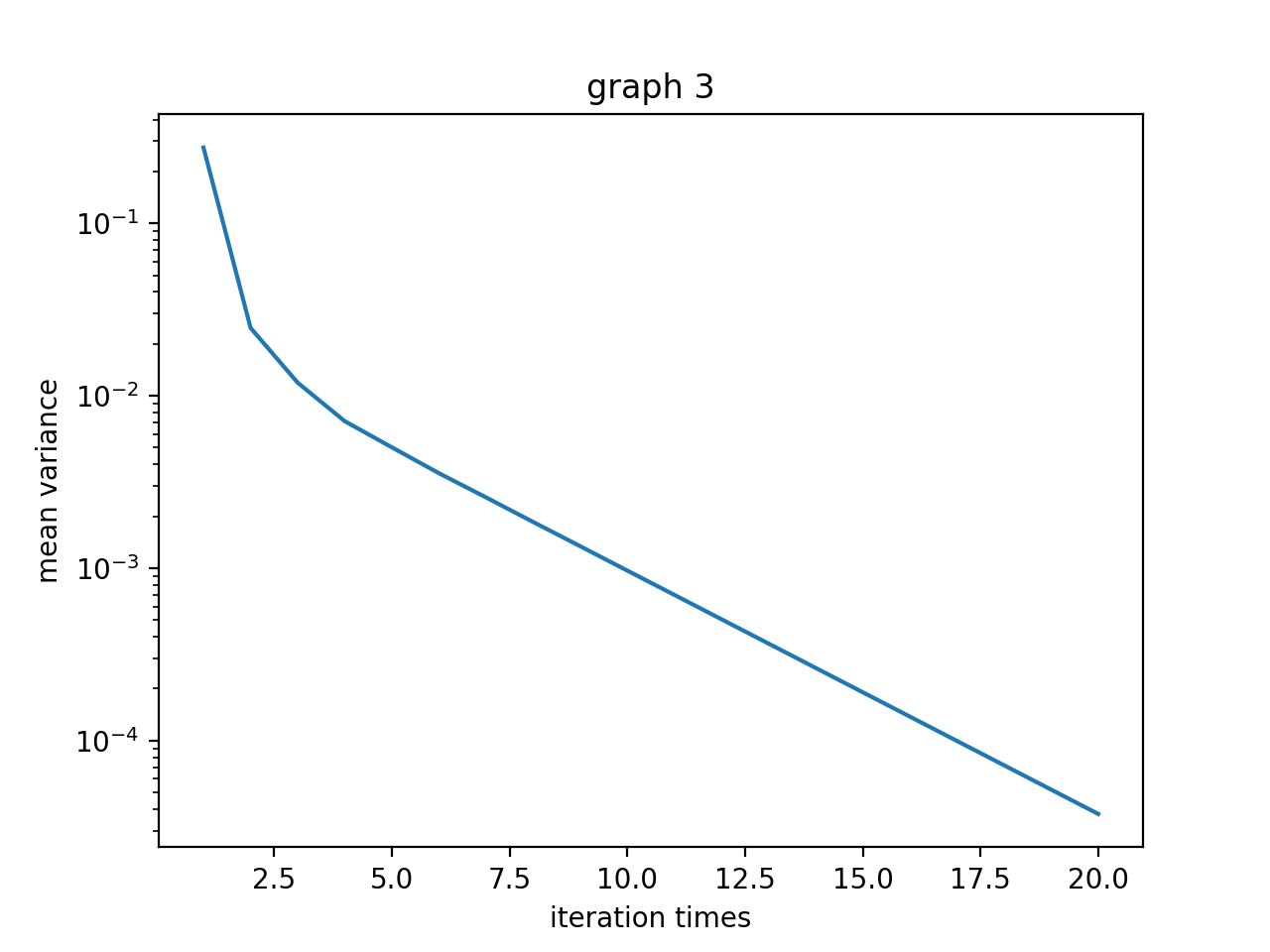
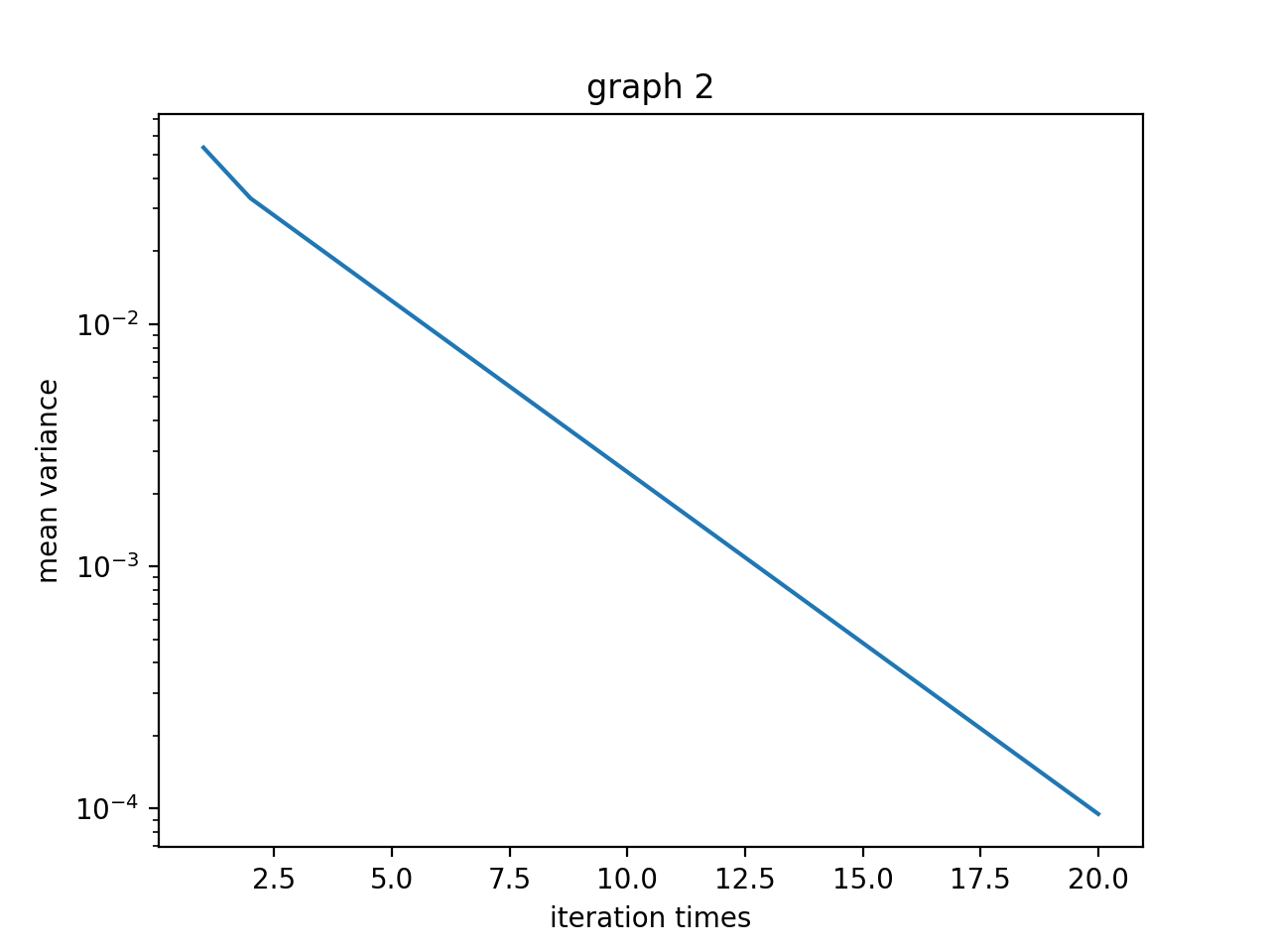
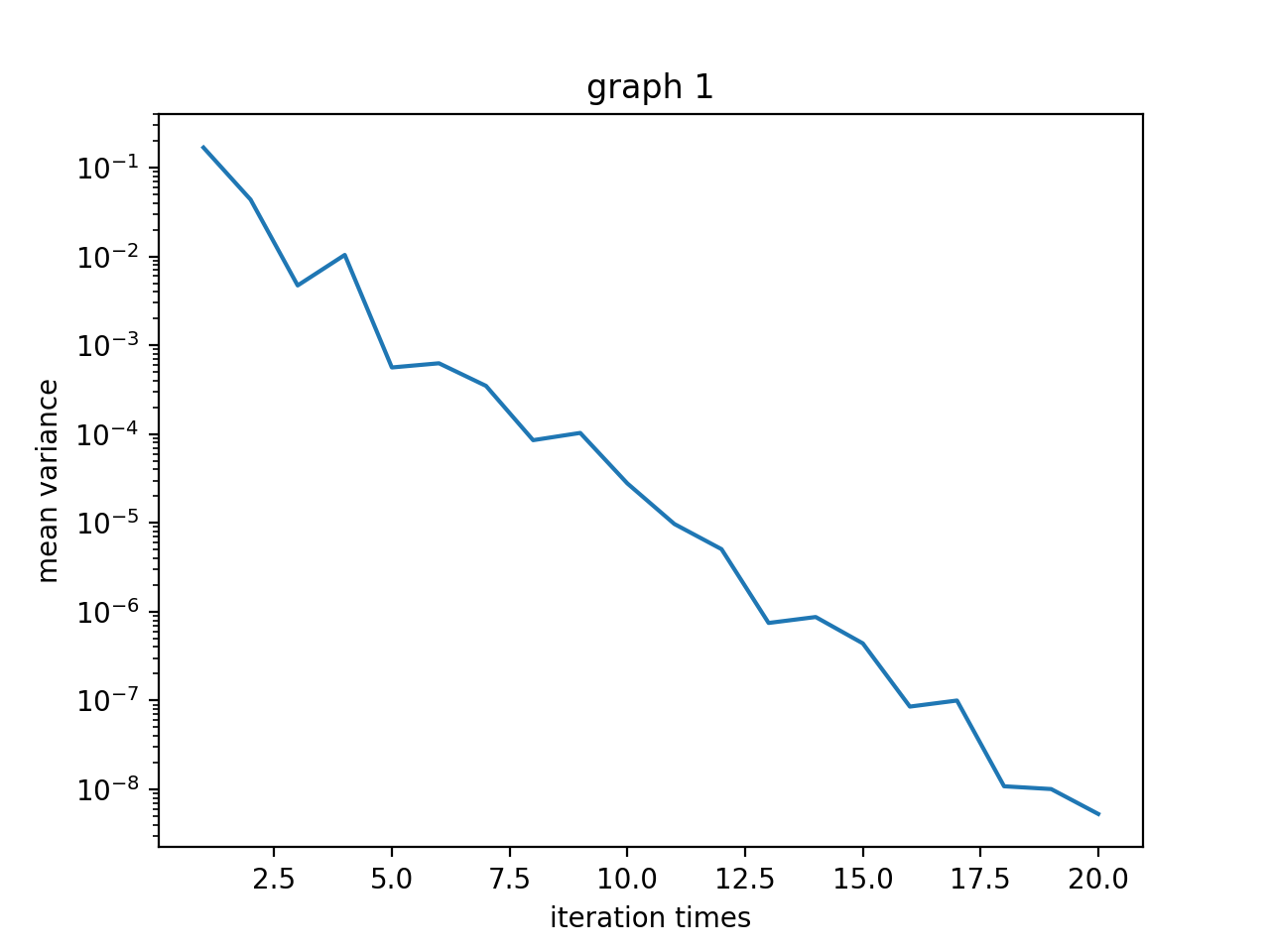
圖片呈現



收斂速度：data1(不收斂)，data2(不收斂)，data3(40次)，data4(68次)，data5(54次)，data6(316次)

二：pageRank

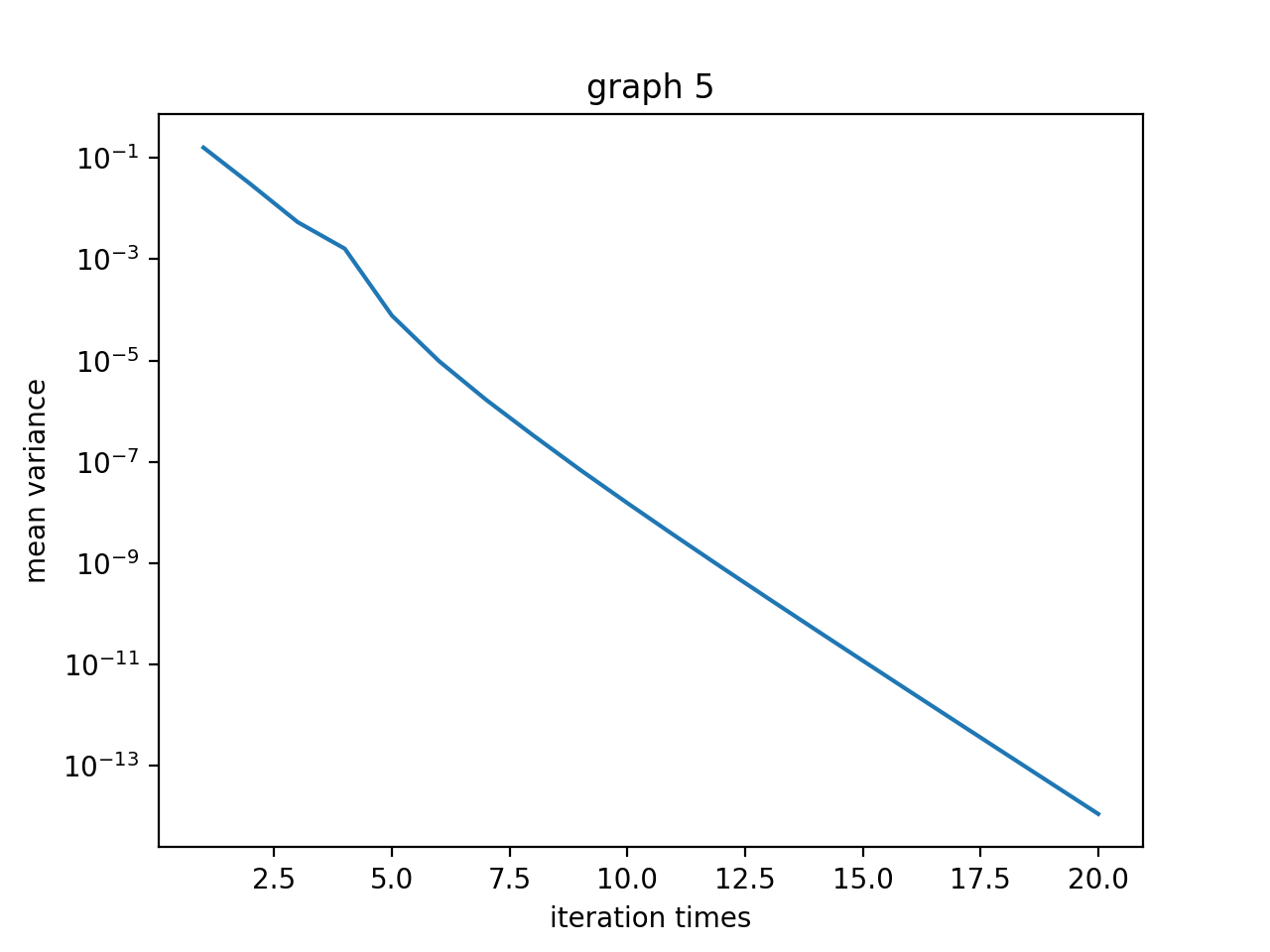
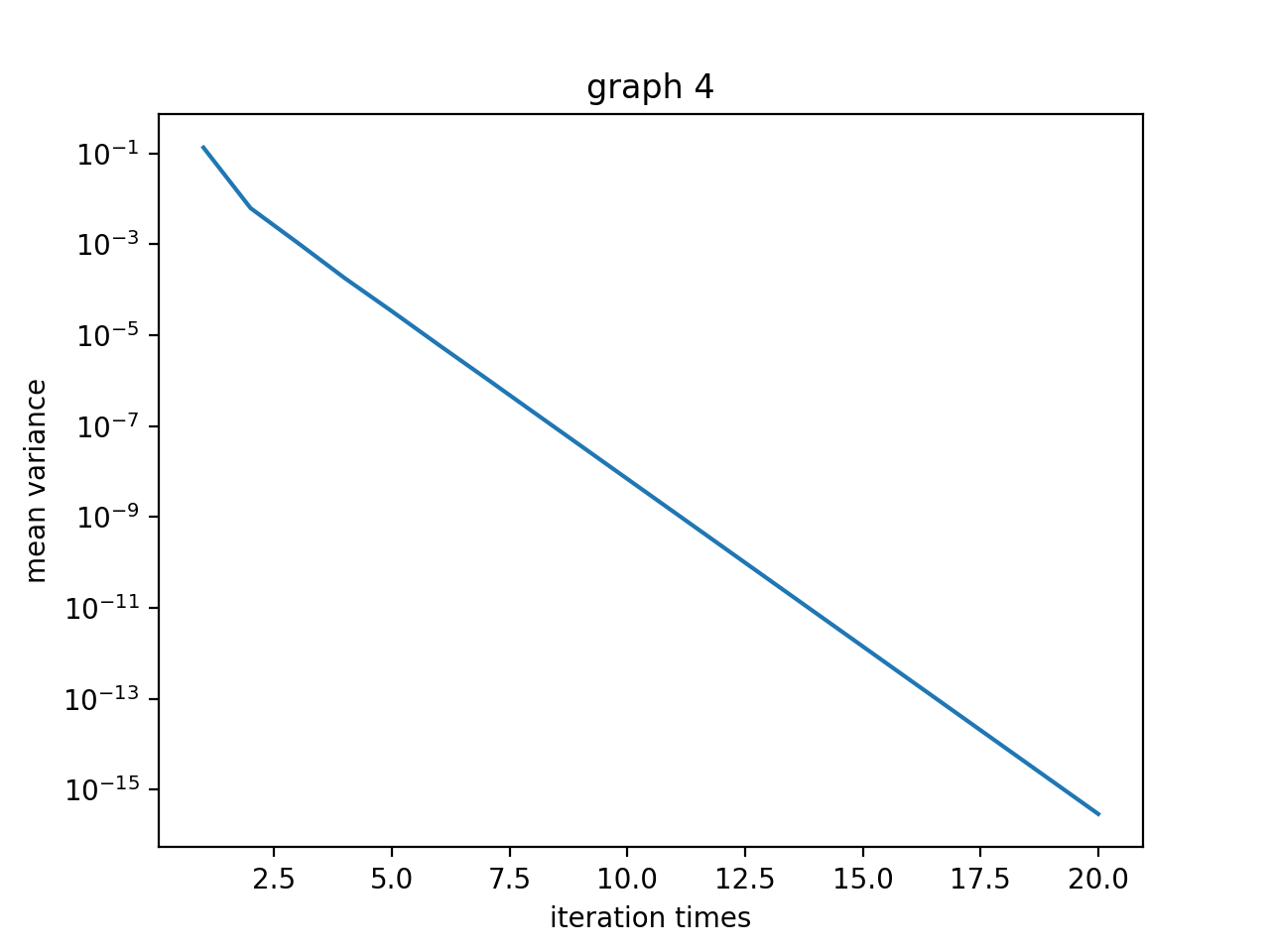
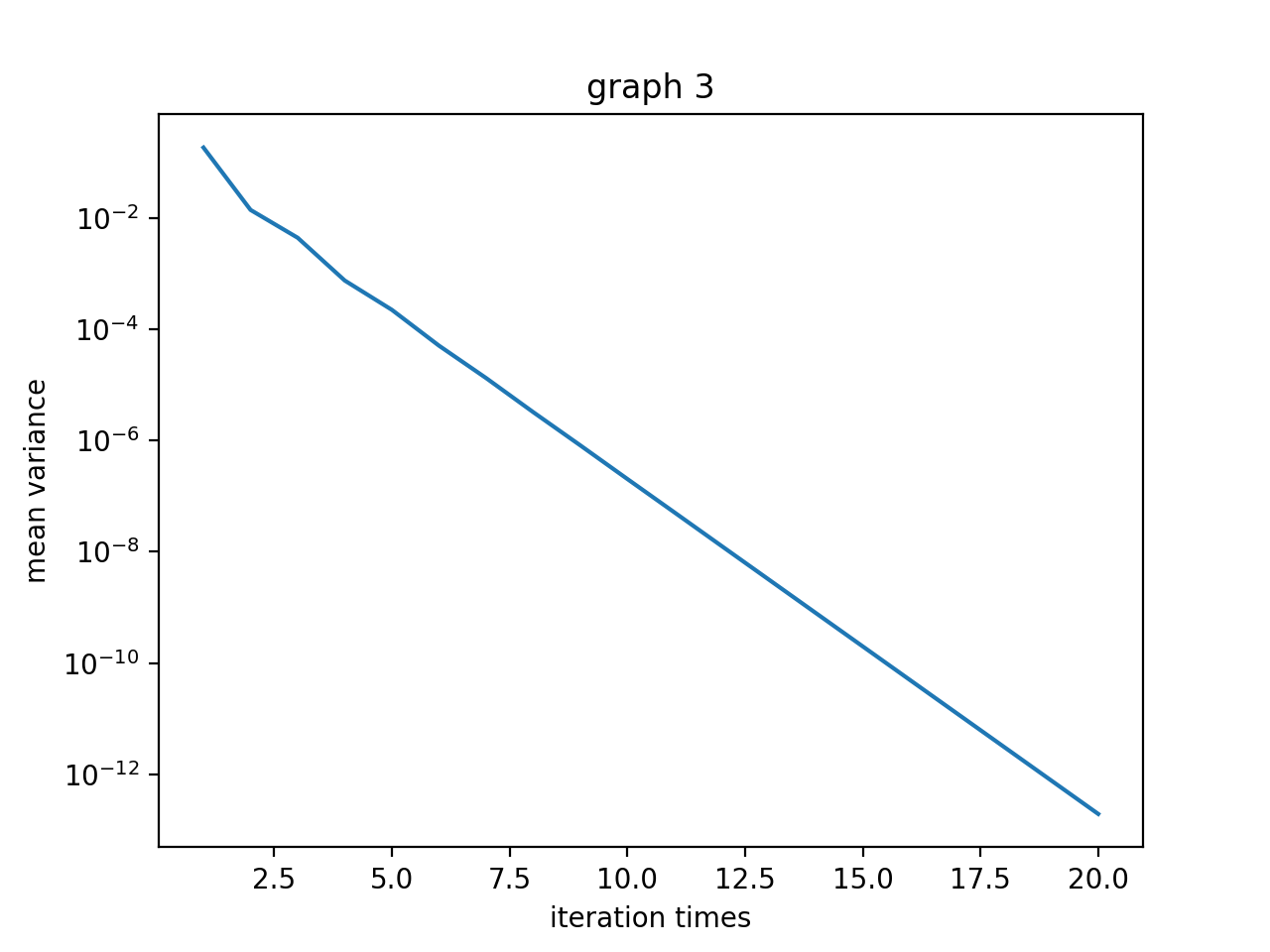
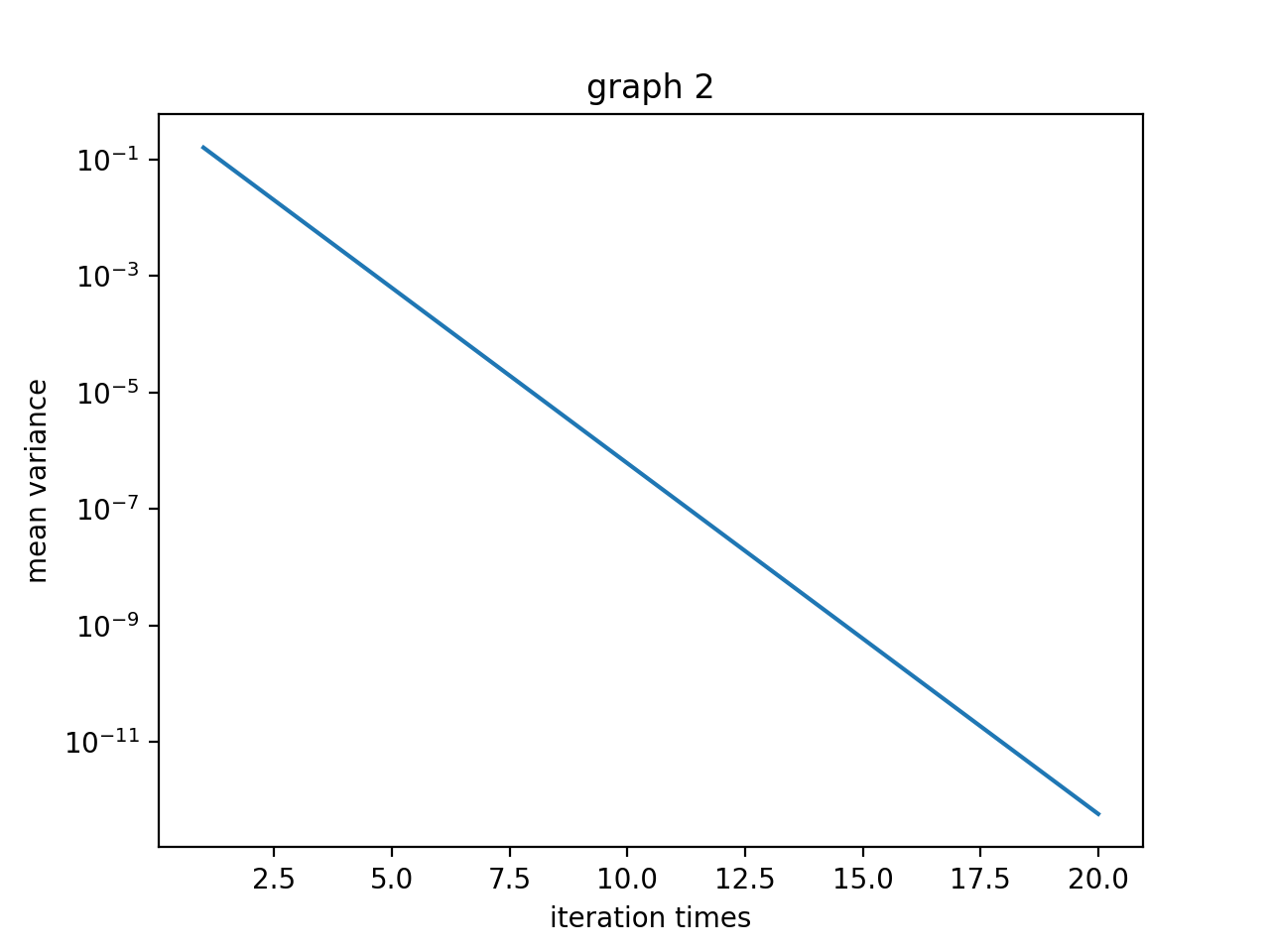
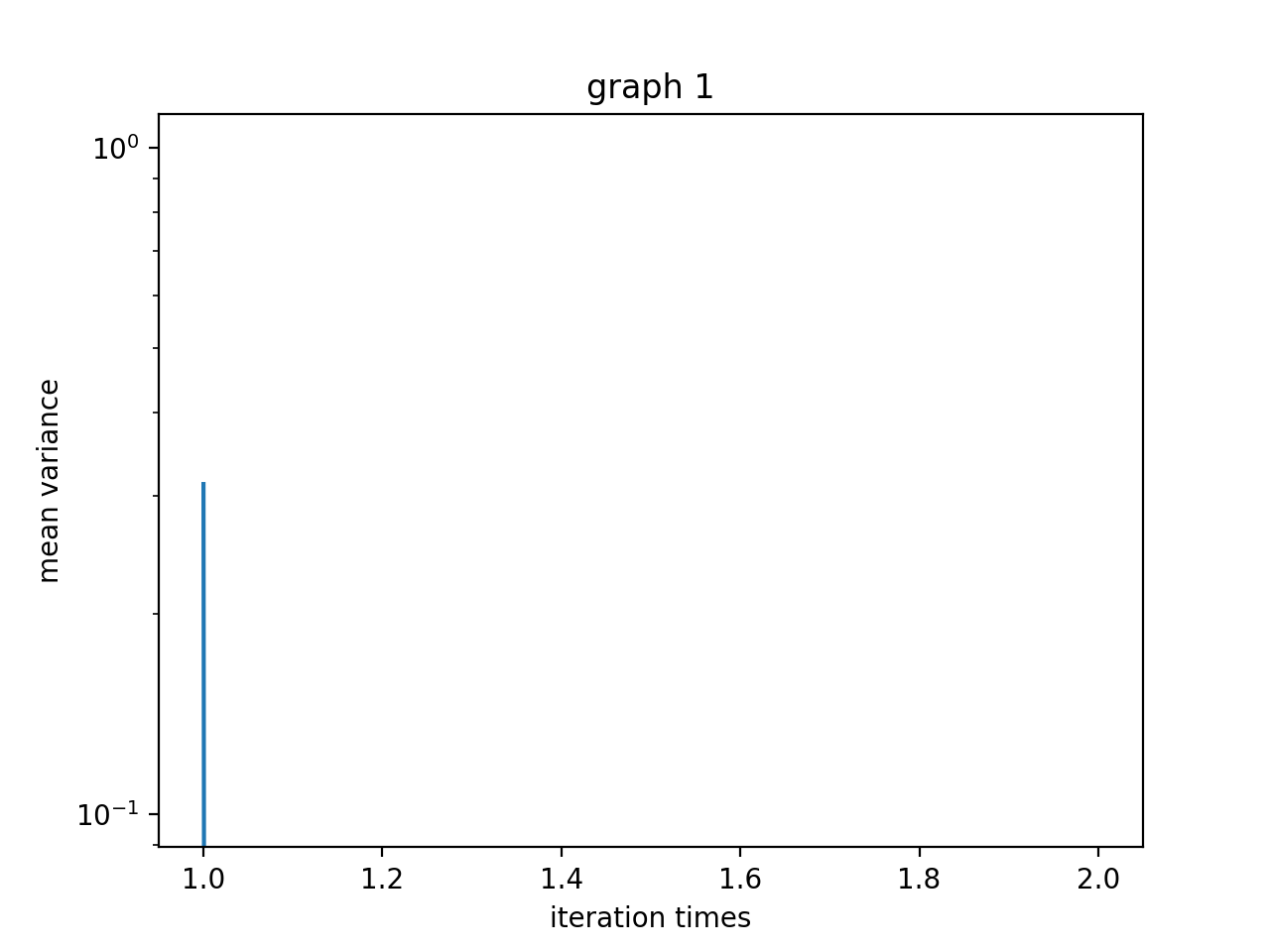
圖片呈現



收斂速度：data1(77次)，data2(203次)，data3(206次)，data4(40次)，data5(145次)，data6(128次)

三：SimRank

圖片呈現



**效能分析**

**Hits**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 圖片1 | 圖片2 | 圖片3 | 圖片4 | 圖片5 | 圖片6 |
| 時間(s) | <0.001 | <0.001 | <0.001 | 0.015 | 0.08 | 0.35 |

**PageRank**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 圖片1 | 圖片2 | 圖片3 | 圖片4 | 圖片5 | 圖片6 |
| 時間(s) | <0.001 | 0.002 | 0.021 | 0.0018 | 0.0629 | 0.28 |

**SimRank**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 圖片1 | 圖片2 | 圖片3 | 圖片4 | 圖片5 | 圖片6 |
| 時間(s) | <0.001 | <0.001 | <0.001 | 0.001 | 12.369 | 無 |

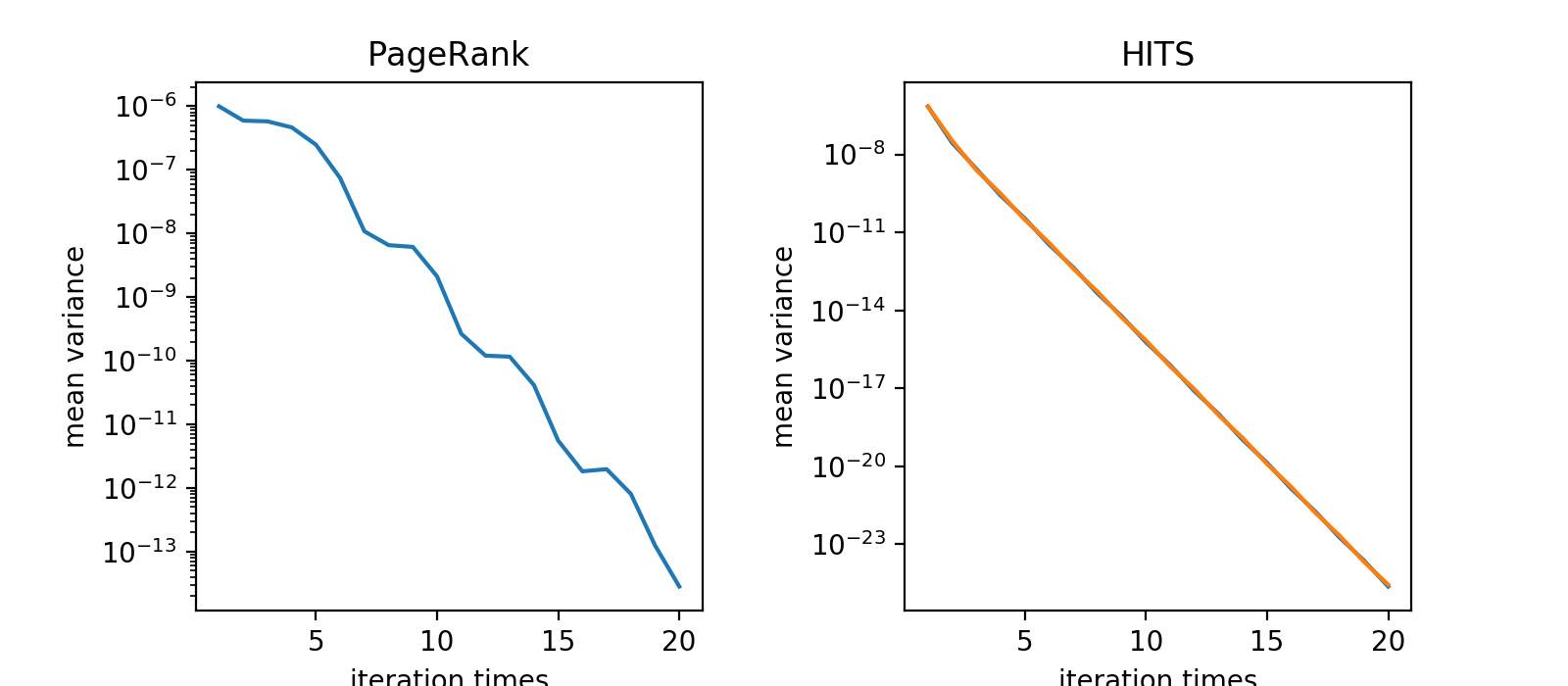
**結論與討論和心得：**

1. pageRank的收斂速度比較快
2. 所以在比較複雜的圖上面，例如data5或data6，pageRank的時間都比較短。
3. pagerank不愧是google崛起的看家演算法，雖然他不是最快效率最好的，但也還堪用，而且他是最早出現的。而page這個名字是來源於google的創辦人Larry Page。
4. 跑simRank的時候，因為遞迴的關係，節點增加時，所需時時間是指數型成長，很花資源。

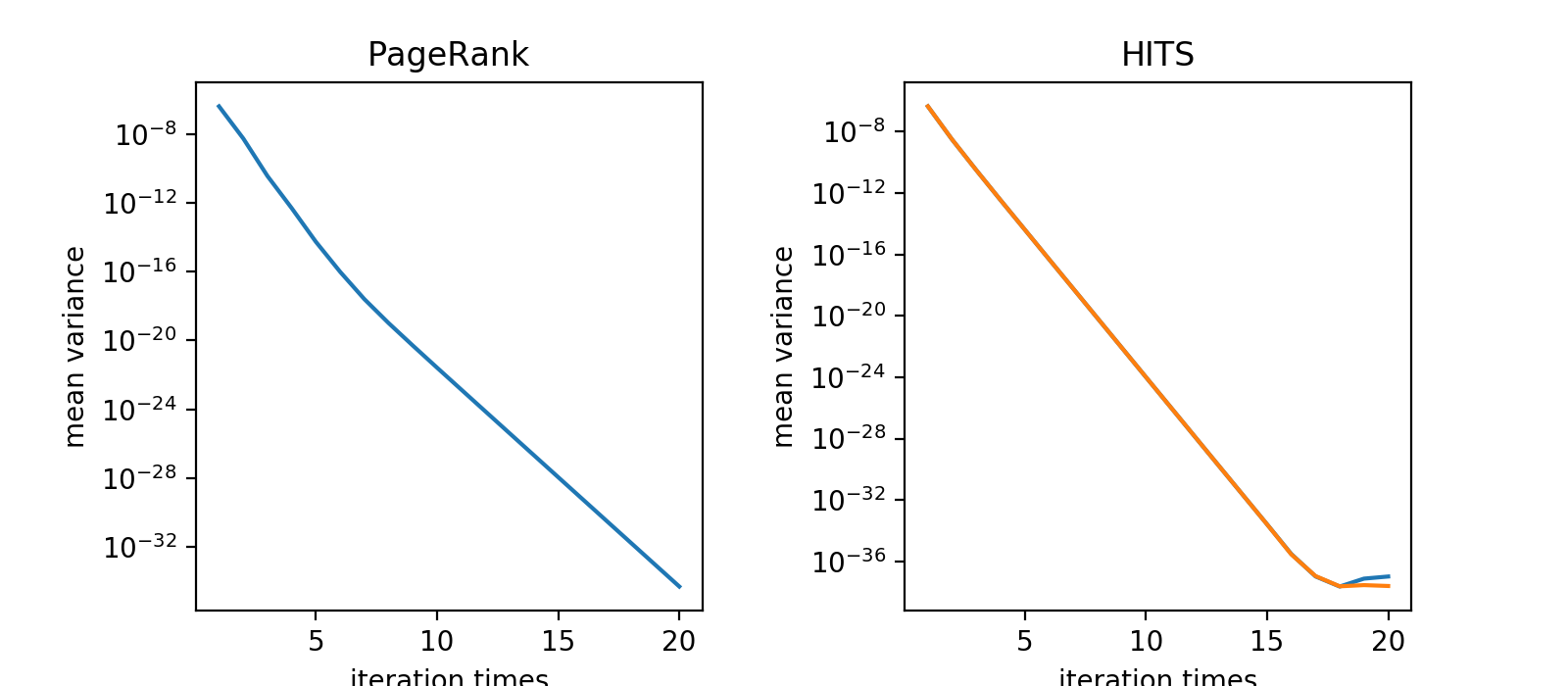


**額外跑project1的data**

**directed**

****

**bidirected**

****

**其他問題討論：**

1. 沒辦法分析會形成loop的網站
2. Can link analysis algorithms really find the “important” pages from Web?

Ans:是的，可以，但要考慮是否有重複點擊同一個網站，導致數值異常的結果。（例如某個網頁的超連結又回到自己行成loop的情況）

1. 要在真實世界時做這個演算法，我覺得最大的問題就是因為真的世界網站數量太多應該是會算不完
2. 不知道這題在問什麼
3. 無
4. C的作用是讓我們決定在迭代的時候參照父母節點的程度為多少，會小就越不像父母節點的相似度，當然會隨著迭代Ｃ的影響力會越來越低。

7. 兩個網站的相似性，我覺得除了可以看他們父母以外，還可以看他們最後是不是來自於同一個祖先節點，如果是，那他們距離同源的祖先節點相隔了幾代，越小的話則相似性越高，可以考慮這個方法計算相似性。