P76074282林品磊 Project1\_report

1. Dataset：
   * data：

1000 transections、number of different item：100

* + data2：  
    10000 transections、number of different item：10000
  + data3：  
    10000 transections、number of different item：100

1. 結果：
   * Apriori + data：  
     + Discussion：Apriori花最多時間的地方在itemset長度=2的時候，因為這時大部分的組合都符合min\_support，所以會產生機乎Item\_size \* Item\_size種結果，尤其在min\_support低於10最明顯。不過之後隨著Itemset長度增加，相對符合的Itemset也減少，計算時間也快了很多。
   * FP-tree + data：  
     + Discussion：FP-tree比Apriori快的原因在於它可以用recursive的去跑，把一個很大、Item又多的資料表recursive成小張的表，然後再把值回傳，就不用每次都要去那張大表找你要的東西的次數。

Run time也直接從140降到0.35整整400倍！

* + FP-tree + data2：  
    - data2秒數比data1多，下降的速度也較慢的原因主要是因為different item增加導致要跑的recursive tree也變多了。
  + FP-tree + data3：  
    - Discussion：data3跟data1的差別只在transections增加了10倍，different item不變，所以兩張的圖形很像，下降的速度也差不多，最大的差別在於因為transections變多了，計算的時間也增加很多。

1. conclusion and observation：

FP-tree跟Apriori的執行速度真的差了好幾百倍，甚至用Apriori去跑data3等到天荒地老、海枯石爛也等不到結果。不過FP-tree所需要的記憶體應該也會比較多，因為在跑recursive的過程要記得前面的東西。如果FP-tree在對每個item去找他們parent item tree的時候可以改成平行處理的話，速度應該還能在上升。

經過這次project我覺得學到很多，主要是更了解了FP-tree跟Apriori。比竟上課了解跟能自己實做出來還有些差別的。