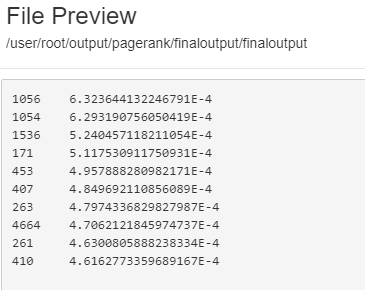
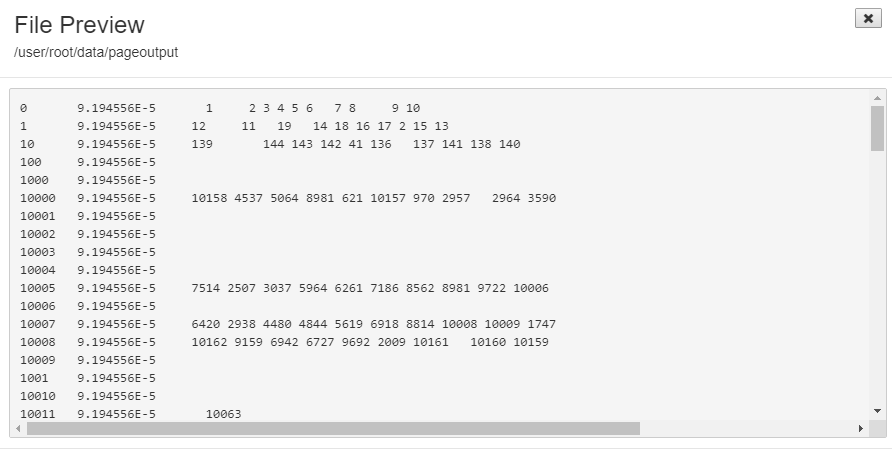
Report

## Final output

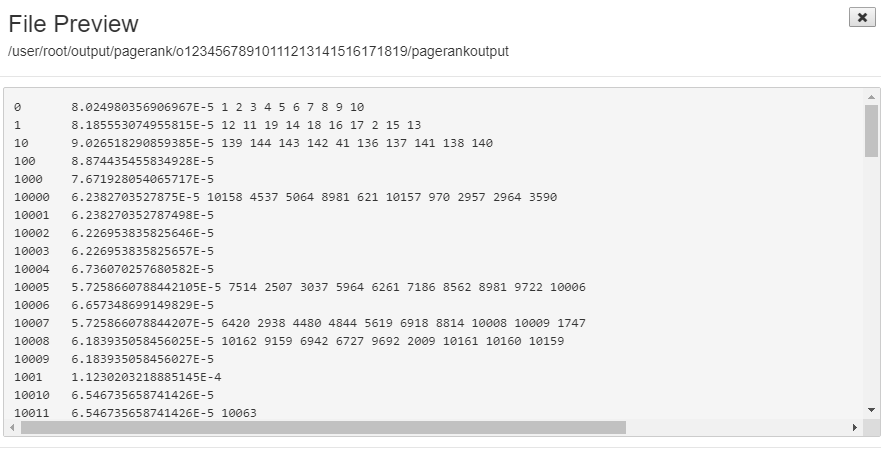
最終結果見finaloutput.txt如下圖：



程式執行過程為先執行Page.java，將給定的p2p-Gnutella04.txt輸入，得到如下圖的結果，並將結果儲存在pageoutput.txt中：



然後執行PageRank.java，將pageoutput.txt輸入，得到如下圖的結果，並將結果儲存在pagerankoutput.txt中：



最後執行TopN.java，將pagerankoutput.txt輸入，得到最終的結果，並將結果儲存在finaloutput.txt中。

## Explain

1. Page.java

此程式的功能是將輸入文本每行的格式:<FromNodeId ToNodeId>如<4 6>

轉換成以下格式，便於之後計算：<FromNodeId PageRank value All ToNodeIds>

如<4 0.2 6 7 8>

Mapper部分輸入<FromNodeId ToNodeId>，把FromNodeId作爲key，ToNodeId作爲value輸出；Reducer部分輸入的key為FromNodeId，value為這個FromNodeId的所有ToNodeId，輸出的key為FromNodeId，value為初始的PageRank value+ All ToNodeIds。

1. PageRank.java

此程式完成PageRank的功能。

Mapper部分輸入格式為： <FromNodeId PageRank value All ToNodeIds>。用id記錄FromNodeId，用double型的pr記錄這個FromNodeId的PageRank值，用count記錄All ToNodeIds的個數，如果count為0，這個點就是dead-ends，並將這個點指向所有點。Mapper部分的輸出分爲兩類，一類輸出的< key ,value>中的key表示其他節點，value 表示當前節點對其他節點的PageRank值，另一類輸出的< key ,value>中的key表示當前節點，value表示其他所有節點。為了區別這兩種輸出，第一類輸出的value裡加入“@”，第二類輸出的value裡加入“&”。

Reducer部分對values中的每一個val進行分析，通過其第一個字元是'@'還是'&'進行判斷是哪一類，並分別處理。如果是'@'，求出當前節點獲得的PageRank值之和，也即是新的PageRank值；如果是'&'，求出當前節點的所有ToNodeIds。然後對新的PageRank值用公式處理，得到最後的結果。最後輸出<FromNodeId PageRank value All ToNodeIds>，與Mapper部分輸入格式相同，這樣就可以進行之後的迭代。

1. TopN.java

此程式的功能是得到10個最大PageRank值的節點，並按題目要求的格式：

<Node PageRank value> 輸出。