

Ricard Meyerhofer Parra**Reyes Vera Capitán****2016-2017 Q1**

CAIM - Fifth Deliverable: Pagerank

Dificultades y elecciones al implementar el código:

Primeramente hemos tenido escoger un lenguaje para la realización de la práctica y nos hemos decantado por Java porque ambos lo conocemos y aunque Python al ser un lenguaje prototipado nos debería de hacer ir mucho más rápido, hemos creído que su poco conocimiento nos haría ir más lentos. Después, hemos tenido que interpretar y comprender cómo funcionan y qué información nos aportan los ficheros de entrada (*airports.txt* y *route.txt*) lo cual no ha sido muy complicado.

Una vez pensadas las estructuras hemos procedido a picar el código propiamente e implementar las funciones de lectura, ordenación y salida donde no hemos tenido mucho problema y posteriormente realizamos la función *computePageRanks* donde en algún momento hemos tenido un error a corregir pues nos sumaba más 1 la suma de P después de cada iteración y también porque queríamos que la implementación fuera lo más eficiente posible (no recorrer de más el Grafo).

Observaciones y comentarios de los resultados:

Aunque nuestra solución quizás pueda ser más eficiente termina realmente rápido por lo cual consideramos que la estructura de datos es adecuada.

Por otra parte, hemos realizado pruebas cambiando el damping por distintos valores donde hemos visto que la variación de este factor nos da lugar a resultados distintos (debido a que el cálculo del vector Q depende del damping factor y por lo tanto, podemos hacer que un aeropuerto con mayor pagerank que tenga una cierta L, pueda tenerlo inferior respecto a otro más pequeño).

También hemos realizado pruebas con la condición de parada del bucle donde se puede observar que nos genera pequeños cambios en nuestra solución pues al comparar dos números si variamos la precisión con la que los comparamos podemos alterar los resultados de esta comparación (que sean iguales dos números que con más precisión son diferentes etc).