Trabajo Práctico Nº2

Informe

75.41 Algoritmos y Programación II   
Cátedra Wachenchauzer  
Cuatrimestre II, 2016

Agustín Manuel Del Torto Herranz. Padrón: 98867.

Víctor Marcelo Belosevich Mugerli. Padrón: 97757

Corrector: Matías Cano

**Comandos:**

**Similares**

a

**Recomendar**

**Camino**

Se solicitó un comando que devuelva un camino entre dos personajes, sugiriendo que se utilice como ruta de comunicación los personajes con los que participaron en mayor cantidad de cómics. Por lo tanto, se implementó una versión modificada del algoritmo de Dijkstra que utiliza esencialmente un **heap de máximos,** prefiriendo así las aristas con mayor peso. Dado que Python no tiene implementado un heap de máximos, se utilizó un heap de mínimos con los pesos invertidos. Al encontrar un camino entre los personajes, devuelve este camino, ya que no es necesario calcular el camino óptimo a todos los vértices.

El tiempo de ejecución de Dijkstra es conocido: ***O(A log V)***

**Centralidad**

c

**Distancias**

c

**Estadísticas**

Ordenes de cada cálculo de estadística:

* Cantidad de vértices: ***O(1).***
* Cantidad de aristas: ***O(1)***. El grafo fue implementado con un atributo contador de cantidad de aristas.
* Grado promedio: ***O(V)***. Se recorren todos los vértices y se suman sus grados.
* Desvío estándar: ***O(V)****.* Se recorren todos los vértices.
* Densidad del grafo: ***O(1)***.

Por lo tanto, el tiempo de ejecución es: ***O(V)***.

**Comunidades**

Para poder determinar las **comunidades** se utilizó un algoritmo de **label propagation**, como se recomendó en la consigna, dado que es simple y funciona como se espera. Al momento de definir la condición de corte, se optó por una cantidad de iteraciones fija (6), dado que el *paper* del algoritmo informa que, en tipo de grafo, para el final de la quinta iteración el 95% de los nodos están etiquetados correctamente.

El tiempo de ejecución es ***O(A)***, siendo A la cantidad de aristas. Para todo vértice i *(O(V))* se visitan sus adyacentes *(O(d(i))* y se busca la etiqueta más frecuente *(O(d(i))*. Por lo tanto: *O(6V\*(2d(V)) = O(V\*d(V)) =* ***O(A).***