

Laboratorio 4: Grafos y hash.

Fecha de entrega: 5 de julio a las 23:59

Stop, por favor Stop.

Tras su largo y extenso esfuerzo de darle una opción para escuchar música a su ser querido, en agradecimiento y premio al buen trabajo le tiene un regalo: más trabajo. Es por esto que se le pide generar una HashTable usando un método bien particular.

- Se le entrega un grafo no dirigido de nombres de canciones (solo el nombre, en serio) como una lista de adyacencia en un archivo CSV. Cada fila tiene el formato: nombre_first,nombre_second,costo,letra.

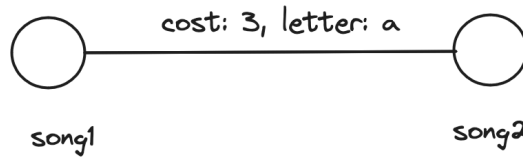


Figure 1: Ejemplo de una fila con los valores: song1,song2,3,a

Consejo: Use un Hashmap de la librería estándar para construir la lista de adyacencia.

- Implemente el algoritmo de Dijkstra para encontrar el camino más corto entre 2 nombres de canciones considerando únicamente el costo. Use el camino más corto para construir una palabra que usará después para el hash. En el siguiente grafo, desde "song1" hasta "song6" el camino más corto genera la palabra "adet".

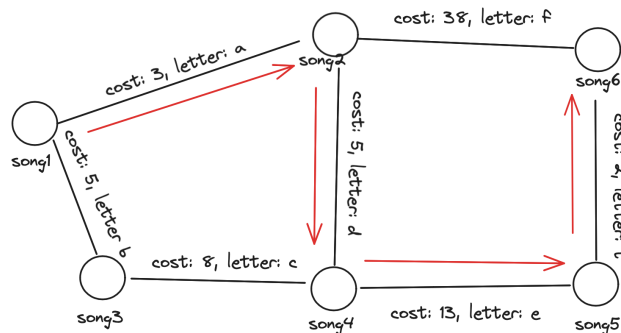


Figure 2: Camino más corto entre song1 y song6, genera palabra "adet"

- Genere una clase HashTable. Esto es un arreglo de alguna estructura que guarde 2 nombres de canciones, tiene tamaño es fijo de 1.000.000 y debe poder guardar elementos calculando el índice que le corresponde a cada par de canciones mediante una función de hash usando la palabra encontrada recorriendo el grafo.
- La forma de generar el índice es aplicar a un string la función

$$\text{hash}(s) = s[0] + s[1] \cdot p + s[2] \cdot p^2 + \dots + s[n-1] \cdot p^{n-1} \bmod m$$

Que corresponde a sumar las letras de un string como valor numérico donde a=1, b=2, c=3 y así hasta z minúscula. Cada caracter se multiplica por una constante p (que es un número coprimo, use $p = 31$ para este laboratorio) y finalmente se calcula módulo m, donde m es el tamaño de la HashTable (en este laboratorio $m = 1.000.000$).

- Finalmente, lea desde otro CSV pares de canciones e inserte en su clase Hashtable los dos nombres (sugerido pero no obligatorio, con una clase Pair que contenga los dos nombres de las canciones) en los índices que le correspondan. Realice esto revisando si hay colisiones al momento de insertar, si existe una colisión muéstrela en pantalla y no inserte el elemento que intentaba insertar. (Una colisión es insertar dos elementos en el mismo índice)