

“Computer science is no more about computers than astronomy is about telescopes.”

Edsger Dijkstra

Distancia musical

Zpotiphy quiere crear un sistema de recomendación basado en los artistas preferidos de los usuarios. Para ello quiere saber la distancia de colaboración entre artistas, la cual se define en base a las relaciones de colaboración entre artistas. Por ejemplo, si un artista X colabora con un artista Y, se dice que la distancia entre Y a X es 1. Continuando con el ejemplo, si un artista Z colabora con el artista Y, se dice que la distancia de Y a Z es 1 y que la distancia entre X a Z es 2, como así mismo si un artista A no tiene colaboraciones con X, Y o Z, se dice que la distancia de A a X es infinity. Por tanto su tarea consiste en ayudar a Spotify escribiendo un programa que devuelva esta información, teniendo en cuenta lo siguiente:

Input: Una línea con el nombre del Artista que se usará como Artista 0. Una segunda línea que contiene dos enteros N y M, que denotan la cantidad de canciones (N) y una cantidad de artistas (M) respectivamente. Las siguientes N líneas traen la base de datos de canciones, cada una de ellas contiene los artistas de la canción y el nombre de esta. Luego vienen M nombres de artistas de los cuales se desea conocer su distancia de colaboración con el Artista 0. **Output:** Una línea por cada artista a consultar y su respectiva distancia con el artista "raíz".

Ejemplo 1:

- Input:
Dave Brubeck
5 3
Stan Getz, Charlie Byrd: Desafinado
Stan Getz, Joao Gilberto: Samba Da Minha Terra
Stan Getz, Dave Brubeck: Get Happy
Dave Brubeck, Paul Desmond: Take the A train
Chon: The space
Dave Brubeck
Charlie Byrd
Chon
- Output:
Dave Brubeck: 0
Charlie Byrd: 2
Chon: Infinity

Ejemplo 2:

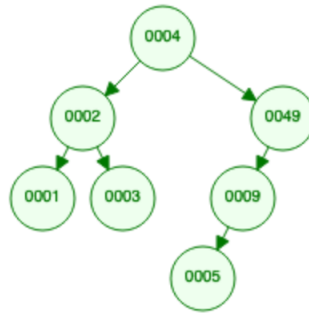
- Input: MF DOOM
3 2
Madvillain, Madlib, MF DOOM, MED: Raid
Rejjie Snow, MF DOOM: Cookie Chips
Jimi Hendrix: Spanish Castle Magic
Jimi Hendrix
Madlib
- Output: Jimi Hendrix: Infinity
Madlib: 1

Hint: Para este problema en específico siéntase libre de utilizar las librerías de Java que estime necesarias. Para llegar a la solución es necesario armar el grafo de los artistas.

Máximo nivel de un BST

Dado el nodo raíz de un árbol de búsqueda binario, compute el nivel del árbol cuya suma de valores sea la máxima. Recuerde que el nivel es la distancia entre un nodo y la raíz del árbol, por lo que los niveles se enumeran desde el 0 para la raíz y van incrementando de uno en uno, hasta llegar a las hojas del árbol.

Por ejemplo sea el siguiente árbol:



Es posible afirmar que el nivel máximo es el nivel 1, dado que $49 + 2 = 51$, lo cual es notablemente mayor que las sumas de los otros niveles.

Implemente un método que reciba la raíz de un BST y compute el nivel cuya suma de valores sea la máxima.

Hint: BFS

Nota: es obligatorio realizar esta implementación utilizando nodos, cualquier implementación que se salte ese paso, se considerará incorrecta.

Torres

Dados N cubos en un orden determinado, su tarea es construir torres de cubos apilando estos uno arriba de otro. Pero existen ciertas restricciones en esta tarea que debe considerar:

- Al momento de apilar un cubo arriba de otro, el cubo a apilar(cubo de arriba) siempre debe ser de menor tamaño que el cubo de abajo.
- Los cubos serán entregados en un orden determinado, para el proceso no es posible reordenar los cubos, estos se deben procesar en el orden que llegan de izquierda a derecha.
- Dadas estas restricciones el objetivo es que calcule la mínima cantidad de torres a generar con los cubos entregados y las restricciones mencionadas.

Input:

- La primera línea contiene un entero N que denota la cantidad de cubos a procesar.
- La siguiente línea contiene N enteros c_i que representan el volumen de cada cubo C_i .

Output:

- Imprimir un entero T , donde T representa la cantidad mínima de torres a generar con los cubos entregados y las restricciones del problema.

Ejemplo:

- Input:
5
3 8 2 1 5
- Output:
2
- La cantidad máxima de torres dado el ejemplo y el orden en que llegan los cubos 3, 8, 2, 1, 5 es 2: la primera torre puede ser compuesta mediante {3, 2, 1} y la segunda torre mediante {8, 5}.

Hint: Para este problema en específico siéntase libre de utilizar las librerías de Java que estime necesarias.

Evaluación:

- Cada problema vale dos puntos, de un total de 6 puntos.
- La copia entre pares se sancionará con la nota mínima.
- La fecha de entrega es para el viernes 25 de Noviembre a las 23:59.
- En CANVAS se creará una entrega para que pueda adjuntar su código.
- Deben trabajar en grupos de dos personas máximo, en caso de trabajar solos notificar al profesor.
- En la entrega deben ir los nombres de los integrantes del grupo, en caso de trabajar en parejas.