

Nombre y apellidos: _____

Firma de comprobación de asistencia al examen:

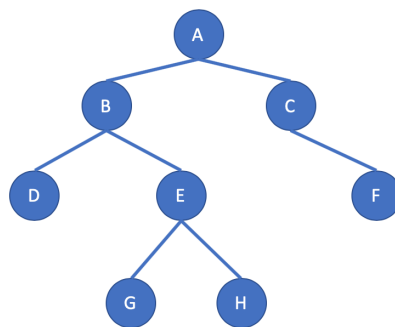
Hora de entrega del examen:

Normas del examen

- La duración del examen es de 2 horas y media.
- Esta parte práctica puntúa sobre 8 puntos. Los otros dos puntos corresponden a las preguntas de teoría.
- En el Aula Virtual está disponible un archivo ZIP que contiene el esqueleto de los ejercicios del examen, así como los tests unitarios correspondientes. Además, contendrá todo el código visto en clase que servirá como apoyo en el desarrollo del examen.
- En todos los ejercicios se valorará la calidad del código implementado en términos de eficiencia, claridad y legibilidad.

Ejercicio 1 [2 puntos]

La clase `LevelsComplete` aumenta la funcionalidad de los árboles binarios para comprobar el nivel de completitud de dichos árboles. En concreto, el método `levelsComplete`, cuyo prototipo está incluido en el código de apoyo del examen, recibe como entrada un árbol binario, y devuelve una colección iterable con los niveles que están completos. Se considera que un nivel está completo si todos los nodos internos de ese nivel tienen dos hijos. Ten en cuenta que se considera que la raíz está en el nivel 1. Por ejemplo, en el árbol siguiente, la colección iterable contendría los valores `{1,3,4}`, ya que los niveles 1, 3 y 4 son completos.



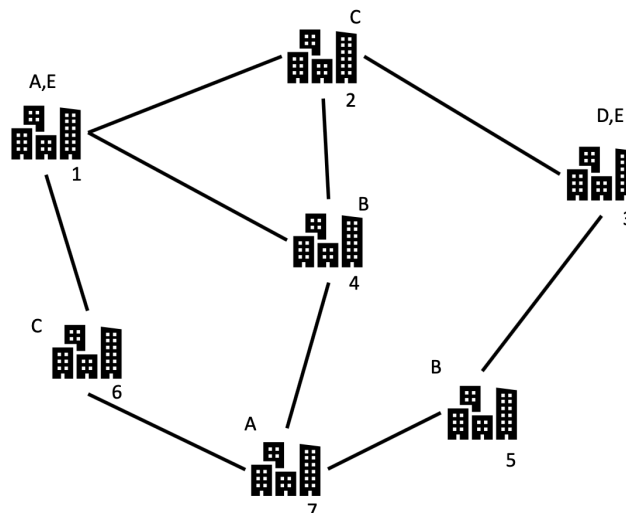
Ejercicio 2 [2 puntos]

Un diccionario es una estructura que permite almacenar bajo una misma clave diferentes valores. Los diccionarios vistos en clase presentan complejidad logarítmica para hacer operaciones de búsqueda. Para mejorar la eficiencia de estas operaciones, se pide implementar la clase `EfficientDict`, que ofrecerá una complejidad de búsqueda de $O(1)$. La clase `EfficientDict`, disponible en el código de apoyo, contiene todos los métodos que se deben implementar con una breve descripción del objetivo de cada uno de los métodos. Se permite la utilización de cualquier

estructura de las vistas en clase (salvo los diccionarios, obviamente). Los tests unitarios ayudarán a comprender el funcionamiento de cada método.

Ejercicio 3 [4.0 puntos]

José es muy alérgico a la leche, corriendo peligro de muerte si no se le atiende rápido. Además, José tiene una tendencia innata a ingerir cualquier producto que contenga leche. Por desgracia, los centros de salud de cada pueblo solo son capaces de atender la alergia de José respecto a unas marcas de leche determinadas. Ahora que ha decidido comprarse una casa, necesita saber en qué pueblo tiene que comprársela para maximizar sus probabilidades de supervivencia. Como a José le gustaría vivir en una zona rural, hay que tener en cuenta que no todos los pueblos disponen de carreteras que los comuniquen entre sí, pero si existe una carretera, será bidireccional.



Por ejemplo, en el conjunto de pueblos que aparecen en la figura anterior, el nombre del pueblo se identifica con un número y las marcas de leche controladas por el centro de salud del pueblo con una letra. Es decir, si José situara su vivienda en el pueblo 4, su centro de salud le salvaría de la marca de leche B, y para salvarse de la marca A debería ir al pueblo 1 o al 7 (ambos están a un salto). Es importante resaltar que José no es tonto, por lo que siempre visitará el centro de salud más cercano que sea capaz de salvar su vida (no se tendrán en cuenta las distancias entre pueblos). Se pide:

- [1 punto]** Definir la estructura de datos necesaria para almacenar todos los pueblos en los que José quiere vivir y las alergias de leche que pueden atender en cada centro de salud. Además, es necesario implementar los métodos `addTown` y `addRoad`, que permitan simular diferentes escenarios.
- [2 puntos]** Implementar el método `milkSafeTown`, que calculará y devolverá cuál es el pueblo que tiene más cerca los centros de salud capaces de salvar a José de cualquier marca de leche. En el ejemplo de la figura, situaría su casa en el pueblo 2, ya que tendría todas las marcas a 1 salto salvo la marca C que la tiene en el propio pueblo (0 saltos). Cualquier otro pueblo le obliga a dar más de un salto, poniendo su vida en riesgo. Ten en cuenta que siempre irá al pueblo que le suponga menos saltos desde su casa.



- c) **[1 punto]** Implementar el método `survive`, que recibirá un pueblo donde José ha bebido leche y el número de pueblos que José puede visitar antes de morir si ha tenido una reacción alérgica. El método devolverá `TRUE` si José visita menos pueblos que el límite para llegar al centro más cercano que pueda tratar su alergia o `FALSE` si José muere porque el camino es demasiado largo.