

PEC 2

EJERCICIO 1 [2 puntos]

Se tiene una cola prioritaria representada con un árbol parcialmente ordenado de aridad 4 (es decir, cada nodo tiene un máximo de 4 hijos). Indique cuantos niveles son necesarios para almacenar 1000 elementos. Diga cuantas comparaciones entre elementos serán necesarias en el peor caso cuando se quiere insertar un nuevo elemento en esta cola de 1000 elementos. Indique cuantas comparaciones entre elementos serán necesarias en el peor caso cuando borramos el menor de los elementos (el más prioritario) de esta cola prioritaria.

El número de nodos x que caben en un árbol de aridad 4 completo de h niveles es $1+4^1+\dots+4^{h-1} = (4^h-1)/(4-1)$, luego despejando:

$$\frac{4^h - 1}{4 - 1} = X; h = \log_4(3x + 1)$$

y como el número de niveles no se puede fraccionar, si el resultado es un número real, hay que redondear al entero inmediatamente superior. De esta manera, el número de niveles necesarios para almacenar 1000 elementos es $\log_4(3001) = 5,77 \rightarrow 6$ niveles.

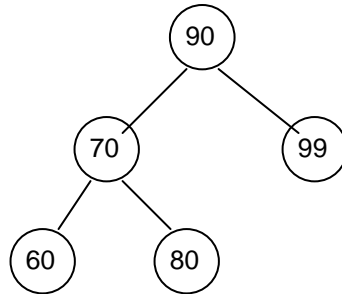
En el peor caso (insertar el elemento más prioritario en un árbol completo, generando un nivel más) tendremos que reestructurar todo el árbol, comparando el nodo insertado (como una hoja en otro nivel) con su padre hasta llegar a la raíz, por lo que se realizarán tantas comparaciones como niveles haya. Este es el caso genérico, pero en nuestro árbol, que almacena 1000 elementos en 6 niveles, queda sitio para otro nodo dentro del nivel 6, por lo que el número de comparaciones necesarias es **5** (no es necesario “abrir” otro nivel, y el número de comparaciones es el número de niveles-1).

Cuando eliminamos el menor elemento (el más prioritario), hay que eliminar el elemento que se encuentra en la raíz, lo que obliga a reestructurar el árbol para que mantenga la relación de orden parcial. En cada reestructuración se debe comparar el nodo con todos sus hijos para intercambiarlo con el menor, y repetir este proceso, en el peor caso hasta alcanzar las hojas, luego hay que comparar $4 * (h - 1) = 4 * 5 = 20$ veces.

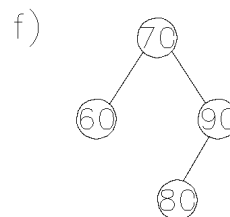
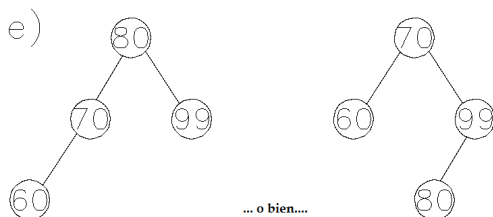
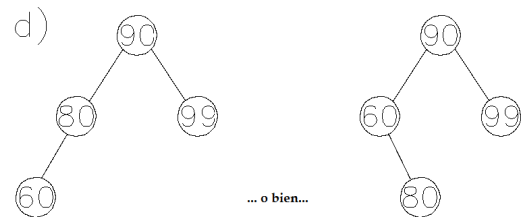
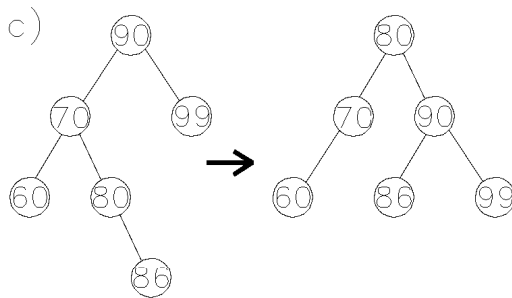
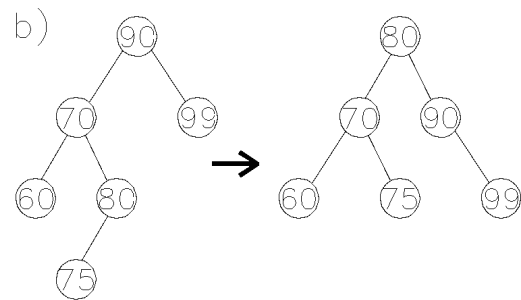
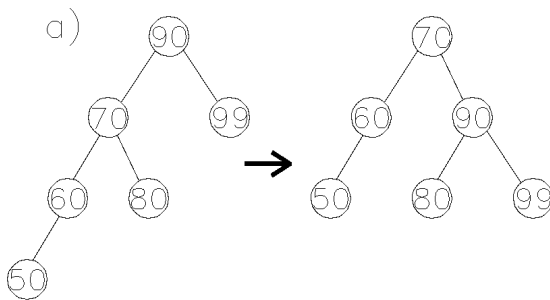


EJERCICIO 2 [2 puntos]

Realizar las siguientes operaciones, según como se ha visto en la asignatura, sobre el árbol AVL de la figura. El resultado de cada operación debe seguir siendo un árbol de búsqueda AVL. Realizar todas las operaciones partiendo del AVL original, que se muestra a continuación:



- Añadir el 50
- Añadir el 75
- Añadir el 86
- Eliminar el 70
- Eliminar el 90
- Eliminar el 99



EJERCICIO 3 [2 puntos]

Suponed que tenemos creado un directorio de enlaces a sitios interesantes de Internet, y queremos añadirle un diccionario de los términos o palabras clave más utilizados en el título y en la descripción de estos sitios.

El diccionario a crear traducirá cada término en una lista encadenada de sitios que tienen ese término como palabra clave. El número de términos a guardar es muy grande e irá creciendo constantemente. Además, es requisito que se pueda recorrer alfabéticamente la lista de términos.

Se pide lo siguiente:

Apartado 3.1) ¿Qué estructura de datos de las estudiadas es la más adecuada para implementar este diccionario? Razonad la respuesta.

Puesto que no se conoce el número de términos que contendrá el diccionario y se puede solicitar un listado ordenado de los mismos, además de que la búsqueda en el mismo se presupone debe ser lo más rápida posible, se podría utilizar un árbol AVL ordenado alfabéticamente donde la clave fuera el término.

Apartado 3.2) ¿Qué modificación se podría hacer a la estructura que permitiera construir un iterador de los términos que empiezan por una letra determinada? Explica brevemente de qué forma se implementaría este iterador.

Se podrían usar las letras del alfabeto como términos almacenados en el AVL (con valor nulo). Estos términos se insertarían todos de golpe al construir el árbol. El iterador a implementar recorrería el árbol AVL en inorden, pero a partir del término correspondiente a la letra. El iterador terminaría su recorrido al encontrar el término correspondiente a la siguiente letra.



EJERCICIO 4 [4 puntos]

El Ministerio de Educación publica varias convocatorias al año para financiar actividades de investigación. Las actividades pueden ser de diversa índole: acciones complementarias, redes de especialización, proyectos de I+D, etc. Un investigador pueden participar simultáneamente en varias de estas actividades, pero solo en un único proyecto de I+D (podría participar en 3 redes y 1 proyecto, por ejemplo). Los investigadores realizan su actividad en el marco de una universidad española, y como fruto de su investigación en la actividad financiada realizan publicaciones científicas.

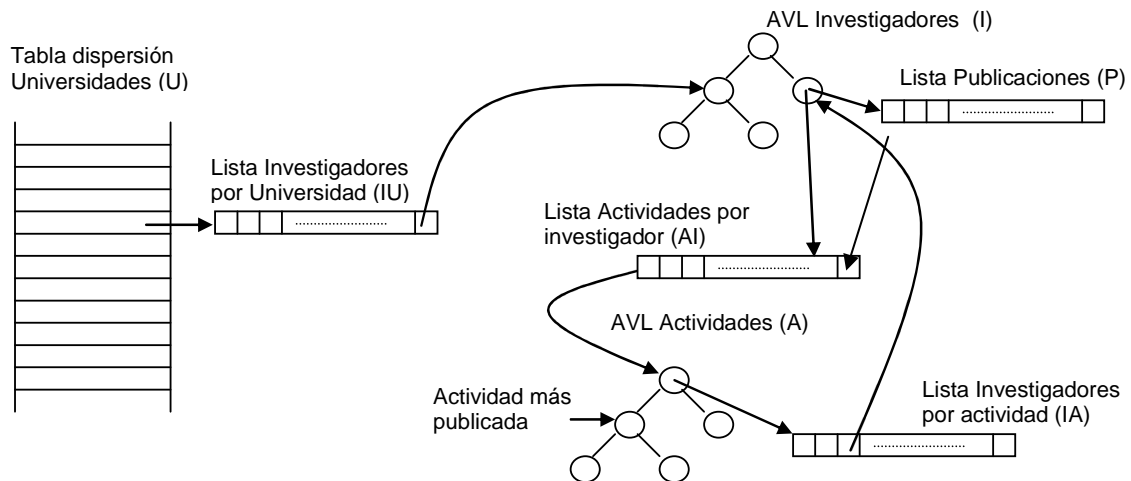
El número de universidades es conocido y estable (llamémosle U). El número de actividades que financia el Ministerio (llamémosle A) es grande, y no se conoce a priori. El número de actividades simultáneas en las que participa un investigador (llamémosle AI) tampoco se conoce, aunque se estima que será pequeño. El número de publicaciones que un investigador puede realizar en todos los proyectos en los que está activo (llamémosle P) es pequeño (del orden de las decenas), pero no tampoco se puede predecir con exactitud. El número de investigadores que participan en una actividad (llamémosle IA) no es fijo, pero si se puede afirmar que no es muy grande, mientras que el número de investigadores (denominado I) de las universidades españolas sí que es grande y variable. Cada universidad tiene un número desconocido de investigadores (llamémosle IU) que depende del tamaño de la misma.

Se necesita diseñar un TAD que refleje esta situación permita realizar las siguientes operaciones:

- `financiarActividad` (lista Investigadores, lista universidades, actividad). Inserta una nueva actividad dentro de las financiadas por el Ministerio. En la actividad participan los investigadores que se pasan como parámetro. Si algún investigador no existe en la estructura, se dará de alta. La universidad a la que pertenece cada investigador figura en la lista que se pasa como segundo parámetro. Si la actividad es un proyecto de I+D, se debe comprobar que no hay entre los investigadores ninguno que participe en otro proyecto de I+D en vigor. Si fuera así, se le excluiría de la lista de investigadores del proyecto. Puede suponerse que las universidades de los investigadores ya existen en la estructura.
- `consultarInvestigadoresUniversidad` (universidad): lista Investigadores. Devuelve los investigadores de una universidad que están inmersos actualmente en alguna actividad de investigación financiada. Puede suponerse que la Universidad siempre existe en la estructura.
- `consultarInvestigadoresActividad` (actividad): lista Investigadores. Devuelve los investigadores que trabajan en la actividad que se pasa como parámetro. Puede suponerse que la actividad siempre existe en la estructura.
- `Publicar` (publicación, investigador, actividad). Da de alta una publicación asociada a una actividad y hecha por un investigador. Puede suponerse que tanto el investigador como la actividad siempre existen en la estructura.
- `MasPublicada()`: actividad. Devuelve la actividad que más veces ha sido publicada.



Se solicita diseñar la estructura de datos que soporte dicha situación y permita realizar las operaciones lo más eficientemente posible en cuanto al tiempo sin malgastar espacio. Realice un dibujo que explique gráficamente la estructura e indique la eficiencia de cada operación especificando las sub-operaciones que se deben realizar para llevarla a cabo, junto con su eficiencia (por ejemplo: insertar en AVL, buscar en lista enlazada, etc).



financiarActividad (lista Investigadores, lista universidades, actividad): $O(\log A + IA + \text{Max}(O(\log I), O(A)))$

Insertar en AVL la actividad: $O(\log A)$

Si la actividad en proyecto de I+D: para cada investigador de la actividad $O(IA)$

 Buscar en AVL investigadores: $O(\log I)$.

 Si no existe

 Insertar en AVL: $O(\log I)$

 Buscar universidad en tabla: $O(1)$

 Insertar investigador en lista de investigadores por universidad: $O(1)$

 Insertar investigador en lista de investigadores por actividad: $O(1)$

 Insertar investigador en lista de actividades investigador: $O(1)$

 Sino

 Comprobar si en la lista de actividades del investigador hay algún I+D

proyecto de vigente: $O(A)$

 Si no tiene proyectos de I+D vigentes

 Insertar investigador en lista de investigadores por actividad: $O(1)$

 Insertar investigador en lista de actividades investigador: $O(1)$

consultarInvestigadoresUniversidad (universidad): lista investigadores: $O(1)$

Buscar universidad en tabla: $O(1)$

Devolver lista asociada: $O(1)$

consultarInvestigadoresActividad (actividad): lista investigadores $O(\log A)$

Buscar actividad en AVL: $O(\log A)$

Devolver lista asociada: $O(1)$



Publicar (publicación, investigador, actividad) $O(\log I + \log A)$

Buscar investigador en AVL: $O(\log I)$

Buscar actividad en AVL: $O(\log A)$

Insertar en lista publicaciones: $O(1)$

Incrementar contador publicaciones actividad: $O(1)$

Si contador actual > contador de la actividad más publicada, actualizar puntero a actividad más publicada. $O(1)$

MasPublicada(): actividad. $O(1)$

Devolver actividad apuntada por puntero a actividad más publicada. $O(1)$

