Ejercicios del Tema 4

1. Implementa una operación en los árboles de búsqueda que *balancee* el árbol. Se permite el uso de estructuras de datos auxiliares.

2. Modifica los diccionarios para permitir almacenar distintos valores para la misma clave. Es decir:

- Al insertar un par (*clave, valor*), si la clave ya se encontrase en el árbol, en lugar de sustituir el valor antiguo por el nuevo, se asociaría el valor adicional con la clave.
- La operación de consulta, en vez de devolver un único valor, devuelve una lista con todos ellos en el mismo orden en el que fueron insertados.
- La operación de borrado elimina todos los valores asociados con la clave dada.

¿Cómo podrías conseguir la misma funcionalidad sin modificar el TAD?

3. Extiende con las siguientes operaciones la implementación de los diccionarios:

- `consultaK`: recibe un entero k y devuelve la k-ésima clave del diccionario (considera que en un diccionario con n elementos las claves se numeran desde k=0 hasta k=n-1)

  ```
  const tClave& consultaK(int k);
  ```

- `recorreRango`: recibe dos claves, a y b, y devuelve la lista con los valores asociados a las claves que están en el intervalo [a, b]

  ```
  Lista<tValor> recorreRango(const tClave& a, const tClave& b);
  ```

- `erase`: recibe un iterador no constante, elimina el elemento apuntado por él y devuelve un iterador al siguiente elemento del diccionario

  ```
  Iterator erase(const Iterator & it);
  ```

¿Cuáles de estas operaciones no varían de una implementación del TAD a otra?

4. En la sección de motivación veíamos que un texto puede venir como una lista de palabras. Otra alternativa es que venga como una lista de líneas, donde cada línea es a su vez una lista de palabras (es decir, el tipo sería `Lista<Lista<string>>`). El problema de las referencias cruzadas consiste en crear el listado en orden alfabético de todas las palabras que aparecen en el texto, indicando, para cada una de las palabras, el número de líneas en las que aparece (si una palabra aparece varias veces en una línea, el número de línea aparecerá repetido). Implementa en C++ un subprogama que reciba un texto y

escriba la lista de palabras ordenada alfabéticamente; cada línea contendrá una palabra seguida de todas las líneas en las que ésta aparece.

¿Qué implementación elegirías para el diccionario?

5. Implementa un TAD `Conjunto` basado en tablas dispersas con las operaciones habituales: `ConjuntoVacio`, `inserta`, `borra`, `esta`, `union`, `interseccion` y `diferencia`.

Y ahora propón una función de codificación adecuada para el TAD `Conjunto`, de manera que sea posible usar conjuntos como claves de una tabla hash.

6. Se define el *índice radial* de una tabla abierta como la longitud del vector por el número de elementos de la lista de colisión más larga. Extiende el TAD `Diccionario` basado en tablas abiertas con un método que devuelva su índice radial.

7. Se llaman vectores dispersos a los vectores para los que el conjunto total de índices posibles es muy grande, y la gran mayoría de los índices tienen asociados un valor por defecto (por ejemplo, cero). Usando esta idea, podemos representar un vector disperso de números reales como un diccionario `Diccionario<int,float>` que sólo almacena las posiciones del vector que no contienen un 0. Implementa funciones que resuelvan la suma y el producto escalar de dos vectores dispersos de números reales.

8. La evaluación continua se le ha ido de las manos al profesor. Les pide a los alumnos que no lo dejen todo para el final sino que vayan estudiando día a día, pero él no predica con el ejemplo. Ahora tiene todos los ejercicios que los alumnos han ido entregando durante todo el año en una pila de folios y le toca revisarlos. Los ejercicios o están bien (y entonces puntúan positivamente) o están mal (y entonces restan). Al final del día quiere tener imprimida una lista con los nombres de todos los alumnos ordenados alfabéticamente y su puntuación en la evaluación continua (resultado de sumar todos los ejercicios que tienen bien menos los que tienen mal). Si un alumno tiene un 0 como balance no debería aparecer en la lista. Aunque sea abusar un poco del alumnado... ¿puedes ayudar al profesor? Lo que se pide es que implementes una función que reciba dos listas de cadenas con los nombres de los alumnos. La primera lista tiene un elemento por cada ejercicio correcto entregado y cada elemento contiene el nombre del alumno que lo entregó (por lo tanto, un mismo alumno puede aparecer varias veces, si entregó varios ejercicios bien). De forma similar, la segunda lista contiene los ejercicios incorrectos. La función debe imprimir por pantalla el resultado final de la evaluación de los alumnos cuyo balance es distinto de 0 por orden alfabético.