

EJERCICIOS EXÁMENES OTROS AÑOS

Ej1. (Feb 19-20).Razonar (en términos de eficiencia de las operaciones) qué TDA elegirías entre los que se indican para:

a) Implementar un **conjunto ordenado de enteros** (con funciones básicas de insertar, borrar, y buscar) de entre: un **ABB** (árbol binario de búsqueda), un **APO** (árbol parcialmente ordenado implementado mediante un **heap**) y un **Vector ordenado**.

EJERCICIOS EXÁMENES OTROS AÑOS

Ej1. (Feb 19-20).Razonar (en términos de eficiencia de las operaciones) qué TDA elegirías entre los que se indican para:

(b) Implementar un **Diccionario** (con funciones básicas de insertar, borrar y buscar) de entre: una **Tabla Hash abierta**, un **vector ordenado** y una **lista ordenada**

EJERCICIOS EXÁMENES OTROS AÑOS

Ej1. (Feb 19-20).Razonar (en términos de eficiencia de las operaciones) qué TDA elegirías entre los que se indican para:

(c) Implementar un **conjunto no ordenado de enteros** (con funciones básicas de insertar, borrar y buscar) de entre: un **AVL**, un **vector no ordenado**, y una **cola con prioridad** (implementada con un heap).

EJERCICIOS EXÁMENES OTROS AÑOS

Ej2. (Feb 19-20). Dadas 2 listas de enteros L1 y L2 implementar una función:

bool check_sum (const list<int> & L1, const list<int> & L2)

que devuelva true si los elementos de L1 pueden agruparse sumando de forma que se puedan obtener los de L2 sin alterar el orden de los elementos. P.ej: Si L1 = {1,2,3,4,1,3,2,5,6,8,3} y L2={ 6, 10, 5, 6, 11} devolvería true

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 3 . (Feb 19-20 Dado un AVL **A**, implementar una función, que, dado un nodo **n** en el árbol, devuelva el nodo siguiente a **n** en el recorrido en inorden

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 4 . (Feb 19-20) Detalla cada una de las operaciones siguientes:

Insertar las claves {8, 16, 12, 41, 10, 62, 27, 65, 13} en una **Tabla Hash cerrada** de tamaño 13. A continuación borrar el 10 y finalmente insertar el valor 51. Resolver las colisiones usando **hashing doble**.

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 5 . (Feb 19-20) Implementar un **iterador** que itere sobre las claves que tengan asociada una lista cuyos elementos sean todos números primos en una clase Diccionario definida como:

```
class Diccionario{  
private:  
map<int, list<int> > datos;  
.....  
.....  
};
```

Han de implementarse (aparte de las de la clase iterador) las funciones begin() y end(). Se supone implementada una función bool primo (int x) que devuelve true si el entero x es primo.

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 5 . (Feb 19-20)

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 6 . (Enero 19-20) Dado el siguiente fragmento de código:

```
{map <int,int> M; M[0]=1; map <int,int> ::iterator p; p=M.find(7);}
```

¿Cual de las siguientes afirmaciones es verdadera?

- a: M no se modifica y p->first=7
- b: M se modifica y p ->first=7
- c: Da un error
- d: M se modifica y p=M.end()

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 7. (Enero 19-20) Se dispone de un vector de gran tamaño donde se almacenan claves de tipo char ordenadas que pueden aparecer de forma consecutiva un numero indefinido de veces.

P.ej: { a,a,a,a,a,b,b,b,c,c,c,c,c,c,c,c,d,d,d,d,d,d,.....}

(a) Implementar un TDA (basado en el tipo map de la STL) capaz de almacenar los datos del vector e implementar una función que dada una posición i del vector original devuelva el valor almacenado en esa posición.

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 7. (Enero 19-20) Se dispone de un vector de gran tamaño donde se almacenan claves de tipo char ordenadas que pueden aparecer de forma consecutiva un numero indefinido de veces.

P.ej: { a,a,a,a,a,b,b,b,c,c,c,c,c,c,c,d,d,d,d,d,d,.....}

(b) Implementar un **iterador** que itere sobre los elementos que cumplan la propiedad de que su número de repeticiones sea un número par. Se deben implementar (aparte de las de la clase iterator) las funciones begin() y end().

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 8. (Enero 19-20) Implementar una función `int minsum(const list<int> &L);` que dada una lista de enteros L, devuelva la suma total de los enteros de la sublista de menor suma.

Ejemplos:

L=[1 -2 3 -3 -1 2 5]; devuelve: -4 (corresponde a la sublista [-3 -1])

L=[1 -2 1 -3 -1 2 5]; devuelve: -5 (corresponde a la sublista [-2 1 -3 -1])

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 9. (Enero 19-20) Implementar una función `int mas_conectado(const vector< set<int> > &VS);` que devuelve el índice `j` tal que el conjunto `VS[j]` es el conjunto que está conectado con un mayor número de otros conjuntos de `VS`. Decimos que dos conjuntos están conectados si no son disjuntos, es decir, si tienen intersección común no vacía. En el caso de haber varios conjuntos con el mismo número de conexiones debe devolver el primero de ellos.

Ejemplos:

`VS=[{0},{1},{2},{0,1,2}]`, devuelve 3

`VS=[{0,1,2},{0},{1},{2}]`, devuelve 0

`VS=[{0,6,9},{5,6,9},{5},{1},{5,9},{5},{1,5,7}]`, devuelve 1

EJERCICIOS EXÁMENES OTROS AÑOS

Ej 10. (Enero 19-20) Insertar (detallando los pasos) las siguientes claves (en el orden indicado):
{47, 31, 49, 66, 50, 52, 82, 38, 7, 63, 53}
en una **tabla hash cerrada** de tamaño 13 con resolución de colisiones usando hashing
doble. Insertar los elementos anteriores en un **APO**. Borrar 1 elemento del APO resultante.