Estructura de Datos. Grupo D

Ejercicios STL. Rosa Mª Rodríguez Sánchez

1. Implementar una función

   int stablepartition (int x, vector<int> & A);

   que, dado un entero **x** y un vector de n enteros **A**, **A= (A0,A1,...,An-1)** reordene el contenido del vector alrededor de un índice **k** de forma que los elementos del subvector **A[0..k]** sean menores o iguales que x, y los elementos del subvector **A[k+1..n-1]** sean mayores que x, y devuelva ese índice k (k=-1 si todos los elementos de A son mayores que x).

2. Una matriz NxN de enteros se dice que está bi-ordenada si cada columna tiene los elementos ordenados de forma no decreciente de arriba abajo, y cada fila tiene los elementos también ordenados de forma no decreciente de izquierda a derecha.
   Ejemplo:

   $$\begin{matrix} 1 & 4 & 9 \\ 9 & 9 & 9 \\ 12 & 14 & 15 \end{matrix}$$

   (a) Implementar una función:

   bool esmatrizbiordenada (vector<vector<int>> & M);

   que devuelva true si M es una matriz bi-ordenada
   (b) Implementar de forma eficiente una función

   bool buscaenmatrizbiordenada (int x, vector<vector<int>> & B);

   que devuelva true si un entero x, está en una matriz bi-ordenada B.
   (c) Implementar un iterador para la clase matriz bi-ordenada, de forma que se recorra por filas y de lugar a una lista de elementos ordenada.

3. Considere el problema de generar todas las subsecuencias ordenadas de la secuencia **X = (1, 2, ..., n).**
   Por ejemplo, si n = 4 las subsecuencias ordenadas de **X = (1, 2, 3, 4)** son: (1), (12), (123), (124), (13), (134), (14) (2), (23), (234) (24), (3), (34) y (4). Esta construcción se puede implementar mediante el uso de una pila S bajo las siguientes reglas:
   - Inicializar la pila con el elemento 1.
   - Si el tope t de la pila verifica t < n entonces apilamos t+ 1.
   - Si t = n, entonces lo desapilamos y, a continuación, si la pila no quedara vacía, incrementamos el nuevo tope de la misma.
   - El algoritmo termina cuando la pila queda vacía.

   Ejemplo:

   $$1\ \begin{matrix} 2 \\ 1 \end{matrix}\ \begin{matrix} 3 \\ 2 \\ 1 \end{matrix}\ \begin{matrix} 4 \\ 3 \\ 2 \\ 1 \end{matrix}\ \begin{matrix} 4 \\ 2 \\ 1 \end{matrix}\ \begin{matrix} 3 \\ 1 \end{matrix}\ \begin{matrix} 4 \\ 1 \end{matrix}\ 2\ \begin{matrix} 3 \\ 2 \end{matrix}\ \begin{matrix} 4 \\ 3 \\ 2 \end{matrix}\ \begin{matrix} 4 \\ 2 \end{matrix}\ 3\ \begin{matrix} 4 \\ 3 \end{matrix}\ 4$$

4. Un tipo de dato pila_doble es un tipo que permite insertar y borrar por ambos extremos. Implementar las operaciones push y pop de este tipo de dato usando como representación un vector. Las operaciones tienen un parámetro que indica la pila sobre la que se hacen las operaciones:
   (a) void pila_doble<T>::push(int numpila, const T & elem)
   (b) void pila_doble<T>::pop(int numpila)

5. Pasar (especificando los pasos seguidos) la siguiente expresión de postfijo a prefijo usando el TDA lineal que creas más apropiado: { a b c + / e f * g h * - - }

6. Dada una pila P de enteros implementar una función

void transformarpila(stack<int> & p);

que transforme una pila p en otra en la que los elementos aparezcan en el mismo orden original y habiendo sido eliminados los elementos que siendo consecutivos aparezcan repetidos.
Ejemplo:
P=<1,1,2,3,3,4,5,5,1,1,9,8,7,7,3> pasaría a quedar como P = <1,2,3,4,5,1,9,8,7,3>

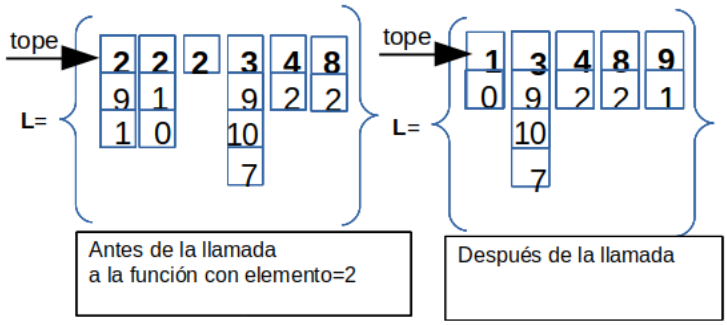
7. Implementar una función

void rotación(queue<int> & C);

que saque una cierta cantidad de enteros del frente de la cola C y los vuelve a insertar al final de cola, de forma que quede en el frente el primer número par que haya en la cola.
Por ejemplo, si C={1,3,5,2,4} ====> C={2,4,1,3,5}

8. Dada una lista que contiene pilas con enteros y que se encuentra ordenada de menor mayor por el tope de cada pila, Implementar una función borrar, que elimina el tope de cada pila en la lista, con valores iguales al dado como parámetro. La cabecera de la función sería:

void Borrar(list<stack<int> > & L, int elemento);



Ejemplo:
Nota: la lista tiene que quedar ordenada por el tope de las pilas trás la ejecución de la función

9. Dadas dos listas, L1 y L2, de tipo T (template) crear una función que obtiene la list intersección (tiene los elementos de L1 y L2 pero no se repiten en caso de que aparezcan el dato tanto en L1 y L2).

10. Usando el TDA list<int>, construir una función

void agrupar_elemento (list<int> & entrada, int k)

que agrupe de forma consecutiva en la lista de entrada todas las apariciones del elemento k en la lista, a partir de la primera ocurrencia. Por ejemplo:
si entrada={1,3,4,1,4} y k = 1, entonces entrada={1,1,3,4,4}, o
si, entrada={3,1,4,1,4,1,1} y k = 1, entonces entrada={3,1,1,1,1,4,4}

11. Dado un vector de listas VL y una lista L, implementar una función:

bool iscomb (vector<list<int>>> VL, list<int> L);

que devuelva true si L es una combinación de las listas de VL en alguna permutación dada. Cada una de las VL[j] debe aparecer una y sólo una vez en L. Por ejemplo: si VL=[(1,2,3),(4,5,6),(7,8)], y L=(7,8,4,5,6,1,2,3) entonces iscomb(VL,L)->true. Pero si L=(3,2,1,7,8,4,5,6)->false.
Ayuda: Escribir una funcion bool isprefix(Lpref,L); que retorna true si la lista Lpref es prefijo de L.Primero chequear que la longitud de L debe ser igual a la suma de las longitudes de los VL[j]. Si pasa este test se procede en forma recursiva. Para cada VL[j] se determina si es prefijo de L. Si ninguna de ellas es prefijo entonces retorna false. Si una (o varias de ellas) lo es entonces se aplica recursivamente iscomb() para determinar si L-VL[j] es una combinación de VL-VL[j]. Es

decir se quita VL[j] del comienzo de L y la posicion j de VL. La recursión se corta cuando L es vacía.

12. Implementar una función:

bool nullsum(list<int> &L, list<int> &L2);

que dada una lista L, devuelve true si hay un rango de iteradores [p,q)) tal que su suma de los elementos en el rango sea 0. En caso afirmativo debe retornar además el rango (uno de ellos, porque puede no ser único) a través de L2. En caso negativo L2 debe quedar vacía. El algoritmo debe ser como mucho cuadrático.
Por ejemplo: si L=(-10, 14, -2, 7, -19, -3, 2, 17, -8, 8) entonces debe retornar true y L2=(14, -2, 7, -19), ó L2=(-8, 8). Si L=(1,3,-5) entonces debe retornar false y L2=().

13. Implementar una función:

void expand(list<int> &L,int m);

que transforme los elementos de una lista L de tal forma que todos los elementos de L resulten ser menores o iguales que m, pero de tal forma que su suma se mantenga inalterada. Para esto, si un elemento x es mayor que m entonces, lo divide en tantas partes como haga falta para satisfacer lacondición; por ejemplo si m=3 podemos dividir a 10 en 3,3,3,1. Es decir si L=(7,2,3,1,4,5), entonces después de hacer expand(L,2) debe quedar L=(2,2,2,1,2,2,1,1,2,2,2,2,1).

14. Implementar una función

void subsecuencia (list<int> &L);

que dada una lista de enteros, elimine todos aquellos que no sean mayores que todos los anteriores. Ejemplo: L={1,3,4,2,4,7,7,1} ==> L={1,3,4,7}

15. Usando la clase set de la STL, construir una función que divida un conjunto de enteros c en dos subconjuntos cpar y cimpar que contienen respectivamente los elementos pares e impares de c y que devuelva true si el número de elementos de cpar es mayor que el de cimpar y false en caso contrario.

16. Dados dos multiset con elementos enteros, implementar la función:

multiset<int> multi_interseccion (const multiset<int> & m1, const multiset<int> & m2)

que calcula la intersección de dos multiset: elementos comunes en los dos multiset repetidos tantasveces como aparezcan en el multiset con menor número de apariciones del elemento.

17. Dado un vector de conjuntos vector< set<int> > VS, implementar una función

bool included(vector< set<int> > &VS);

que devuelve true si los conjuntos del vector VS son subconjuntos propios en forma consecutiva, es decir, si Sj está incluido en Sj+1 para j = 0, ..., n-2. Debe devolver false en caso contrario.

18. Dadas dos map, M1 y M2, definidos como:

map<string,int> M1,M2;

siendo el primer campo el nombre de una persona y el segundo campo al número de seguidores. Obtener el mapa correspondiente a la unión en el que el número de seguidores será la suma de los seguidores en M1 y los seguidores en M2 para esa misma persona que aparece en M1.

19. Dado un map<string,int> M queremos saber:
    - El número de elementos en el map que hay entre dos claves en este caso de tipo string. Por ejemplo el número de elemento almacenados entre las entradas [patata,zanahoria].
    - El número de elementos en el map que tiene como información asociada un valor en un rango,por ejemplo entre [20,100].

20. Implementar una función:

void apply_map(const list<int> &L, map<int,int> &M, list<int> &ML);

que, dada una lista L y un map M devuelve en ML una lista con los resultados de aplicar M a los elementos de L. Si algún elemento de L no está en el dominio de M entonces el elemento correspondiente de ML no es incluido. No pueden usarse estructuras auxiliares. El tiempo de ejecución del algoritmo debe ser O(n), donde n es el número de elementos en la lista (asumiendo que las operaciones usadas del map son O(1)).
Por ejemplo, si L = (1, 2, 3, 4, 5, 6, 7, 1, 2, 3) y M = {(1, 2), (2, 3), (3, 4), (4, 5), (7, 8)} entonces después de hacer apply_map(L,M,ML), debe quedar ML = (2, 3, 4, 5, 8, 2, 3, 4).

21. Se desea construir un traductor de un idioma origen a un idioma destino. Una palabra en el idioma origen puede tener más de un traducción en el idioma destino.Dar una representación para el TDA Traductor usando el tipo <map>.
Implementar la función insertar que añade una palabra del idioma origen junto con las traducciones en el idioma destino. Implementar la función consultar que obtiene las traducciones de una palabra en el idioma destino. Implementar la clase iteradora dentro de la clase Traductor para poder iterar sobre todas las palabras.

22. Implementar un iterador que itere sobre las claves que sean números primos en una clase Diccionario definida como:

```
1 class Diccionario{
2 private:
3 map<int, list<string> > datos;
4 ..........
5 ..........
6 };
```

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

23. Se dispone de un conjunto de claves de gran tamaño, que se almacenan de forma ordenada. La particularidad de este conjunto es que las claves pueden aparecer de forma consecutiva un número indefinido de veces.
Por ej: { 1,1,1,1,1,3,3,3,6,6,6,6,6,6,6,9,9,9,9,9,.....}
Para el acceso a las claves se pretende diseñar un TDA con la misma especificación del vector de la STL, sin embargo se busca una representación basada en el tipo <map> que almacene los datos con un espacio de memoria proporcional al número de claves distintas existentes. ¿Qué representación propondrías? Considerando dicha representación, construye una función que dada una posición i del vector original devuelva el valor almacenado en esa posición.