UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Estructuras de datos. Curso 2021-2022
Convocatoria ordinaria de Enero.
Grado en Ingeniería Informática.
Doble Grado en Ingeniería Informática y Matemáticas
Doble Grado en Ingeniería Informática y ADE

3. (1 punto) Implementa una función:

   void resume_cola(list<int> &L,queue<int> &Q);

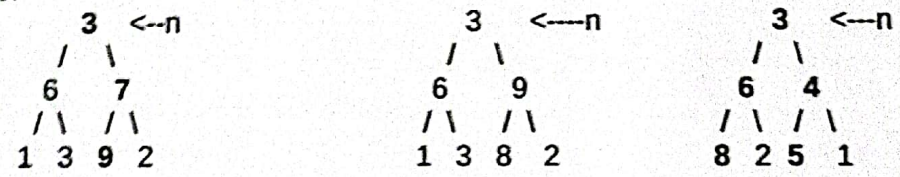
   que va tomando un elemento entero n de la cola Q y, si es estrictamente positivo, saca n elementos de L (si ya no quedan n elementos, saca todos los que queden) y los reemplaza por su producto. Si el elemento de la cola es negativo o cero, no hace nada. Esto ocurre con todos los elementos de Q hasta que se acaben, o bien se acaben los elementos de L. **No pueden usarse estructuras auxiliares.**
   Por ejemplo: Si L=(**1,3,2**,1,4,**5,3**,2,4,1) y Q=(**3**,2,-1,0,**2**,5,2,-3) entonces L debe quedar así: L=(**6**,4,**15**,8), y Q=(2,-3) (es decir, sobran elementos de Q).
   Otro ejemplo: Si L=(**1,3,2**,1,4,5,**3,2,4,1,3,2**,1,4,7) y Q=(**3**,2,-1,0,2,**5**) entonces L debe quedar así L=(**6**,4,15,**48**,1,4,7), y Q=() (es decir, sobran elementos de L que quedan como estaban en la lista)

4. (1 punto) Dado un árbol binario A y un nodo n de ese árbol, implementa una función:

   bool secuencia_creciente (const bintree<int> &A, bintree<int>::node n);

   que devuelva true si existe algún camino desde n a una hoja en la que se cumpla que cada etiqueta de un nodo hijo tiene un valor mayor que la del nodo

   Ejemplo:



   true **false** **true**

5. (1 punto) Dado un **vector de conjuntos** vector<set<int> > V, implementa una función

   void esta_en_conjunto(vector<set<int> > &v, map<int,list<int> > &recuento);

   que devuelva a través del map **recuento** cada uno de los elementos que aparecen en algún conjunto y una lista de las posiciones del vector en las que se puede encontrar.

   Por ejemplo, si v=[{1,20,3}, {20,3,45}, {3,45,5,93}, {20,45,6,8}, {93}, {1,3,5}] entonces debe devolver

   recuento={<**1**,{0,5}>, <**20**, {0, 1,3}>, <**3**, {0,1,2,5}>, ... , <**5**, {2,5}>, ... }

6. (1 punto)

   A) Inserta (detallando los pasos) las siguientes claves (insertadas en ese orden) en un **AVL**: {2, 1, 4, 5, 9, 3, 6, 7} especificando los pasos seguidos e indicando cuando sea necesario el **tipo de rotación** que se usa para equilibrar y mantener la estructura de AVL

   B) Inserta (detallando los pasos) las siguientes claves (en el orden indicado):

   {51, 35, 53, 70 ,54 ,56, 86, 42, 11, 67, 57}

   en una **tabla hash cerrada** de tamaño 13 con resolución de colisiones usando hashing doble.

Tiempo: 2.30 horas

UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Estructuras de datos. Curso 2021-2022
Convocatoria ordinaria de Enero.
Grado en Ingeniería Informática.
Doble Grado en Ingeniería Informática y Matemáticas
Doble Grado en Ingeniería Informática y ADE

1. (0.75 puntos)

   A) Dadas las 3 siguientes afirmaciones:

   (a) En un esquema de hashing doble puede usarse como función hash secundaria

   $$h_0(k)=[\ M-(k\%M)\ -1\ ]\%M, \text{ con M primo.}$$

   (b) Es correcto hacer la siguiente definición **set <stack <int> > :: iterator q;**

   (c) Un **APO** puede reconstruirse de forma unívoca dado su recorrido en preorden.

   Indica la respuesta correcta (**Razonando la respuesta**):

   (a) Las tres son ciertas. **(b)** Dos son ciertas y una falsa **(c)** Dos son falsas y una cierta

   (d) Las tres son falsas.

   B) Si busco el elemento máximo en un APO con el mínimo en la raíz:

   (a) El elemento debe estar necesariamente en una hoja.

   (b) El algoritmo para encontrarlo es O(n)

   Indica la respuesta correcta (**Razonando la respuesta**):

   (a) Las dos afirmaciones son ciertas **(b)** Una es cierta y la otra falsa **(c)** Las dos son falsas.

   C) Supongamos que hacemos las 2 siguientes afirmaciones:

   (a) En un árbol binario el número de nodos con 2 hijos es igual al número de hojas menos uno.

   (b) Para cualquier valor k, la estructura final de un AVL que se construye insertando los enteros 1,2,..., 2^{k-1}, en ese orden, es la de un árbol completo.

   Indica la respuesta correcta (**Razonando la respuesta**):

   (a) Las dos afirmaciones son ciertas **(b)** Una es cierta y la otra falsa **(c)** Las dos son falsas.

2. (1.25 puntos) Tenemos una clase **libro** que permite gestionar las palabras de un libro. Para ello usa un **map<string, list<pair<int,int> > >** de forma que para cada palabra hay una lista **list<pair<int,int> >** dónde cada par contiene un número de capítulo y la posición dentro del mismo en la que aparece dicha palabra.

   A) Implementa un método **convertir_vector** que devuelva **vector<list<string> >** de forma que en la posición **i** contenga todas las palabras del capítulo **i+1** ordenadas alfabéticamente y sin repeticiones. Por ejemplo, si tenemos el map:

   <informática, {<**1**,10>,<**2**,10>,<**3**,41>}>
   <telemática, {<**1**,2>,<**1**,21>,<**2**,50>,<**3**,31> }>
   <robótica, {<**3**,30>,<**4**,5>}>

   el vector contendría:
   v[0]={informática,telemática,telemática} ->**v[0]={informática,telemática}**
   v[1]={informática, telemática}
   v[2]={informática,robótica,telemática}
   v[3]={robótica}

   B) Implementa una clase **iterator** que itere sobre las palabras del libro que aparezcan en capítulos impares y en posiciones pares de ese capítulo. Implementa los métodos de la clase iterator y los métodos **begin()** y **end()** de la la clase libro.