

Estructuras de datos

Grados de la Facultad de Informática (UCM)

27 de mayo de 2022

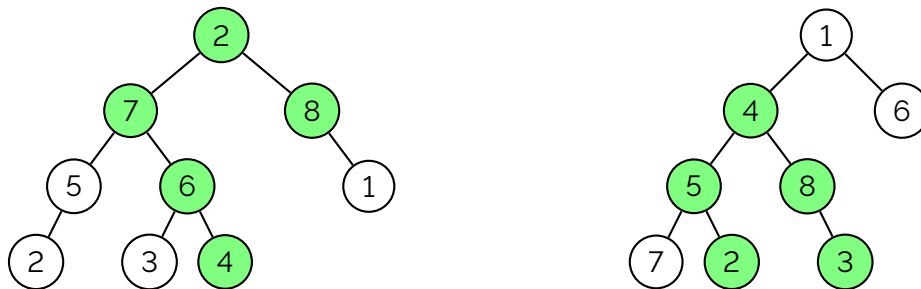
Normas de realización del examen

1. El examen dura **3 horas**.
2. Debes desarrollar e implementar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc>.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen.
4. Escribe tu **nombre y apellidos** en el espacio reservado para ello en los ficheros proporcionados.
5. Dispones de un fichero plantilla para cada ejercicio. Debes utilizarlo atendiendo a las instrucciones que se dan en cada fichero y escribiendo tu solución en los espacios reservados para ello, siempre entre las etiquetas `<answer>` y `</answer>`.
6. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.
7. Si necesitas consultar la documentación de C++, está disponible en <http://exacrc/cppreference>.

Ejercicio 1 (2 puntos)

Definimos un *camino* entre dos nodos de un árbol binario como una secuencia de nodos $n_1 n_2 \dots n_k$ sin repeticiones (por cada nodo del árbol se pasa como mucho una vez) tal que para todo par de nodos consecutivos $n_i n_{i+1}$ ($1 \leq i < k$) uno de ellos siempre es padre del otro (n_i es padre de n_{i+1} o n_{i+1} es padre de n_i).

Dado un árbol binario de números enteros positivos nos queremos fijar en el camino que tiene la mayor cantidad de números pares. En particular, queremos saber esa cantidad máxima. Por ejemplo, en los siguientes árboles están marcados caminos con una cantidad máxima de números pares. En el árbol de la izquierda el camino tiene 4 pares y no hay ningún camino con más. En el árbol de la derecha el camino con más pares tiene 3. El camino señalado es uno de ellos, pero hay otros como el 2 5 4 8, o el 6 1 4 8 3 (todos con 3 pares).



Implementa una función que reciba un árbol de la clase `BinTree` y calcule la máxima cantidad de números pares que se pueden encontrar en un camino. Justifica el coste de esta función.

Entrada

La entrada comienza con el número de casos que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario: si el árbol es vacío se representa con un `-1`; si no, primero aparece su raíz, y a continuación la descripción del hijo izquierdo y después la del hijo derecho, dadas de la misma manera.

Salida

Para cada caso, se escribirá la máxima cantidad de números pares que se puede encontrar en un camino.

Entrada de ejemplo

```
4
2 7 5 2 -1 -1 -1 6 3 -1 -1 4 -1 -1 8 -1 1 -1 -1
1 4 5 7 -1 -1 2 -1 -1 8 -1 3 -1 -1 6 -1 -1
1 3 -1 -1 5 -1 7 -1 -1
1 2 -1 -1 -1
```

Salida de ejemplo

4	
3	
0	
1	

Ejercicio 2 (2 puntos)

Queremos implementar una función que reciba una lista y la modifique para que tenga solamente la primera aparición de cada elemento diferente.

```
template<typename T>
void eliminar_repetidos(std::list<T> & l);
```

Por ejemplo, si la lista es $l = [1, 5, 1, 1, 2, 5, 2]$ entonces `eliminar_repetidos(l)` haría que la lista se transformara en $l = [1, 5, 2]$. Es decir, solamente se conserva la primera aparición de cada valor, y estos mantienen su orden de aparición.

Implementa esta función, que debe manipular la lista de forma abstracta, es decir, sin acceso a su representación interna.

El coste de la operación debe ser lineal con respecto al número de elementos en la lista. Justifica el coste de tu implementación.

Entrada

Cada caso de prueba está formado por tres líneas. La primera contendrá el carácter N si los elementos de la lista son números, o el carácter P si los elementos son palabras. La segunda línea contendrá un entero $L > 0$ que representa la longitud de la lista. La tercera línea contendrá los L elementos de la lista. Si son números estarán en el rango $[0..10^9]$. Si son palabras, estarán formadas por no más de 30 caracteres de la 'a' a la 'z'.

Salida

Para cada caso de prueba se escribirá una línea con los elementos de la lista después de eliminar los elementos repetidos.

Entrada de ejemplo

```
N
7
1 5 1 1 2 5 2
P
6
perro gato perro gato perro gata
P
1
uno
```

Salida de ejemplo

```
1 5 2
perro gato gata
uno
```

Ejercicio 3 (3 puntos)

Queremos implementar un TAD `GaleriaArte` que simule el comportamiento de una galería de arte simplificada. En una galería de arte se exponen los obras de diversos artistas. En cualquier momento la galería puede aceptar exponer nuevas obras, o puede vender alguna de las que tiene expuestas. Se hará uso de los tipos `artista` y `obra` que se representan con un `string`.

Las operaciones de la galería de arte son las siguientes:

- `nueva_obra(c,a,p)`: añade a la galería una nueva obra `c`, del artista `a`, con un precio `p`. Si la obra `c` ya estaba en la galería, se lanza una excepción del tipo `invalid_argument` con el mensaje `Obra ya en la galeria`.
- `venta_obra(c)`: simula la venta de la obra `c`, lo que incluye que se elimina de la galería de arte. Si la obra no está en la galería se lanzará una excepción del tipo `invalid_argument` con el mensaje `Obra no existente`.
- `mas_vendidos()`: devuelve un vector con todos los artistas cuyo total de ventas (suma de los precios de las obras que se han vendido de ese artista) es el máximo en la galería, ordenados por orden alfabético de menor a mayor.
- `obras_por_antiguedad(k)`: devuelve en un vector las k obras que más tiempo han permanecido en la galería sin venderse, ordenadas desde la que más tiempo lleva a la que menos. Si no hay k obras en la galería se devolverán todas ellas.

La implementación de las operaciones debe ser lo más eficiente posible. Por tanto, debes elegir una representación adecuada para el TAD, implementar las operaciones y justificar la complejidad resultante.

Los métodos del TAD no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso está formado por una serie de líneas, en las que se muestran las operaciones a llevar a cabo, una por cada línea: el nombre de la operación seguido de sus argumentos. La palabra `FIN` en una línea indica el final de cada caso.

Los nombres de los artistas y obras de arte no tienen blancos.

Salida

Para cada caso de prueba se escribirán los datos que se piden. Las operaciones que generan salida son:

- `mas_vendidos`, escribe la cabecera `Los artistas que mas han vendido son:` seguido del nombre de los artistas uno en cada línea.
- `obras_por_antiguedad`, escribe la cabecera `Obras mas antiguas en la galeria:`. En las líneas siguientes se escribe el nombre de las k obras que más tiempo llevan en la galería una por línea. Si no hay k obras en la galería se escribirán todas ellas.

Si alguna operación produce una excepción se mostrará el mensaje `ERROR:` seguido del mensaje de la excepción como resultado de la operación, y nada más.

Cada caso termina con una línea con tres guiones (`---`).

Entrada de ejemplo

```
nueva_obra Autorretrato Goya 150
nueva_obra Las_meninas Velazquez 90
nueva_obra La_metamorfosis Dali 45
nueva_obra Fiesta_en_Figueras Dali 45
venta_obra La_metamorfosis
nueva_obra La_venus_del_espejo Velazquez 120
venta_obra Fiesta_en_Figueras
mas_vendidos
obras_por_antiguedad 2
venta_obra Las_meninas
mas_vendidos
nueva_obra Biarritz Sorolla 80
nueva_obra El_pie_herido Sorolla 50
venta_obra Autorretrato
mas_vendidos
nueva_obra La_rendicion_de_Breda Velazquez 130
nueva_obra Cosiendo_la_vela Sorolla 70
nueva_obra Aquelarre Goya 100
venta_obra El_pie_herido
nueva_obra La_metamorfosis Dali 90
obras_por_antiguedad 10
venta_obra La_metamorfosis
mas_vendidos
FIN
nueva_obra Autorretrato Goya 150
venta_obra Las_meninas
mas_vendidos
FIN
```

Salida de ejemplo

Los artistas que mas han vendido son:

Dali

Obras mas antiguas en la galeria:

Autorretrato

Las_meninas

Los artistas que mas han vendido son:

Dali

Velazquez

Los artistas que mas han vendido son:

Goya

Obras mas antiguas en la galeria:

La_venus_del_espejo

Biarritz

La_rendicion_de_Breda

Cosiendo_la_vela

Aquelarre

La_metamorfosis

Los artistas que mas han vendido son:

Dali

ERROR: Obra no existente

Los artistas que mas han vendido son:
