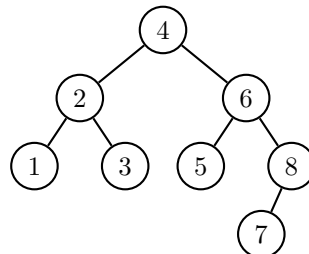


Reconstrucción de un árbol binario de búsqueda

Este semestre nos ha tocado diseñar bonitos árboles binarios de búsqueda, con números enteros en los nodos. En estos árboles la raíz contiene un valor que es estrictamente mayor que todos los del hijo izquierdo y estrictamente menor que todos los del hijo derecho y, además, ambos hijos son también árboles binarios de búsqueda. Aquí tenemos un ejemplo:



Cuando hemos acabado con ellos, para evitar problemas también los hemos guardado en dos archivos, en uno serializados según su recorrido en preorden (raíz, hijo izquierdo, hijo derecho) y en el otro serializados según su recorrido en inorden (hijo izquierdo, raíz, hijo derecho). Pero ha ocurrido un desastre y hemos perdido el segundo fichero. ¿Tendremos que volver a empezar a diseñar árboles? Aunque suene muy divertido, en realidad no lo es tanto. ¿Podremos reconstruir los árboles originales a partir solamente de su recorrido en preorden?

Requisitos de implementación.

Se implementará una función que reciba el recorrido en preorden de un árbol binario de búsqueda (por ejemplo en un **vector**) y devuelva un objeto de la clase **bintree** que represente un árbol que tenga ese recorrido.

Entrada

La entrada está formada por una serie de casos. Cada caso ocupa una línea y contiene el recorrido en preorden de un árbol binario de búsqueda (cuyos nodos tenían valores entre 1 y 10^6).

Salida

Para cada caso, se escribirá el recorrido en postorden del árbol reconstruido.

Entrada de ejemplo

```
4 2 1 3 6 5 8 7
2 4 6
12 8 10 20
```

Salida de ejemplo

```
1 3 2 5 7 8 6 4
6 4 2
10 8 20 12
```

Autor: Alberto Verdejo.