

VIDEO EXPLICATIVO VIDEO

CONSIGNA Escribir una función `void path2tree(list<list<int>> &paths, int root, tree<int> &T);` que, dada una lista de caminos `paths` reconstruye el árbol `T` a partir de la raíz `root`.

EJEMPLOS

| | | | |
|---|---------------------|--------------------|----------------------------------|
| <code>paths: [[0,1,2], [1,3]],</code> | <code>root:0</code> | <code>=></code> | <code>T:(0 (1 2 3))</code> |
| <code>paths: [[4,3,1], [3,2], [4,0,5]],</code> | <code>root:4</code> | <code>=></code> | <code>T:(4 (3 1 2) (0 5))</code> |
| <code>paths: [[3,5,1,0], [1,4], [5,2]],</code> | <code>root:3</code> | <code>=></code> | <code>T:(3 (5 (1 0 4) 2))</code> |
| <code>paths: [[1,2,3], [1,0,4], [0,5]],</code> | <code>root:1</code> | <code>=></code> | <code>T:(1 (2 3) (0 4 5))</code> |
| <code>paths: [[2,0,4], [0,1], [2,5], [2,3]],</code> | <code>root:2</code> | <code>=></code> | <code>T:(2 (0 4 1) 5 3)</code> |

AYUDA Primero veamos como insertar un camino `L` en el ‘arbol, después vemos como armar el árbol a partir de todos los caminos.

- Buscar con `find` el nodo `n` que contiene el primer elemento de `L` en `T`
- Para cada elemento `x` de `L` recorrer los hijos `c` de `n` hasta encontrar `x`.
- Si `x` no está entre los hijos, entonces insertarlo al final.
- Continuar con el siguiente elemento de `L` a partir de ese nodo `c` (ya sea que el valor no estaba o que fue insertado).

Como armar todo el árbol:

- Insertar `root` en la raíz de `T`.
- Para cada camino `L` buscar (con `find()`) el valor del primer elemento de `L` en `T` y agregar el camino a partir de allí.

NOTAS

- Los valores nodales son únicos, es decir que al hacer un `find` el nodo retornado es único.
- Los valores nodales pueden estar repetidos en los caminos. Por ejemplo si pasamos los caminos `[[0,1,2], [0,1,3]]` (el 0 y el 1 están en los dos caminos).
- Los caminos están ordenados apropiadamente, es decir que cuando vamos a insertar el camino `L`, el primer elemento de `L` está garantizado que ya está insertado en `T`.

ZIP Enlace al zip