

Mira esa secuencia... ¿Es un vector? ¿Es una lista?

¡No! ¡¡Es un Súper Árbol!!

Martín Knoblauch Revuelta

<http://www.mkrevuelta.com> @mkrevuelta mkrevuelta@gmail.com

indizen  using std::cpp

Except where otherwise noted, this work is licensed under:  
<http://creativecommons.org/licenses/by-nc-sa/4.0/>



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

Presentación disponible en mi blog semiabandonado:

<http://www.mkrevuelta.com>

# Índice

1. El problema
2. Super Árbol
3. Vista no proporcional
4. Aplicaciones
5. Propuestas similares
6. Reflexionemos

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Introducción al problema

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Are lists evil?—Bjarne Stroustrup



<https://isocpp.org/blog/2014/06/stroustrup-lists>

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

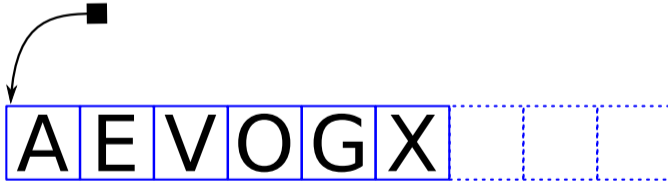
Aplicaciones

Propuestas  
similares

Reflexionemos

# Array

- Acceso aleatorio rápido
- Inserción/extracción... lentas



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

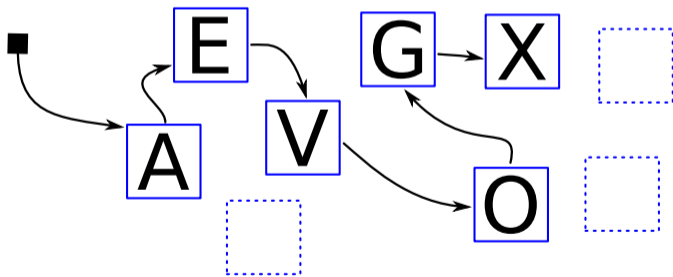
Aplicaciones

Propuestas  
similares

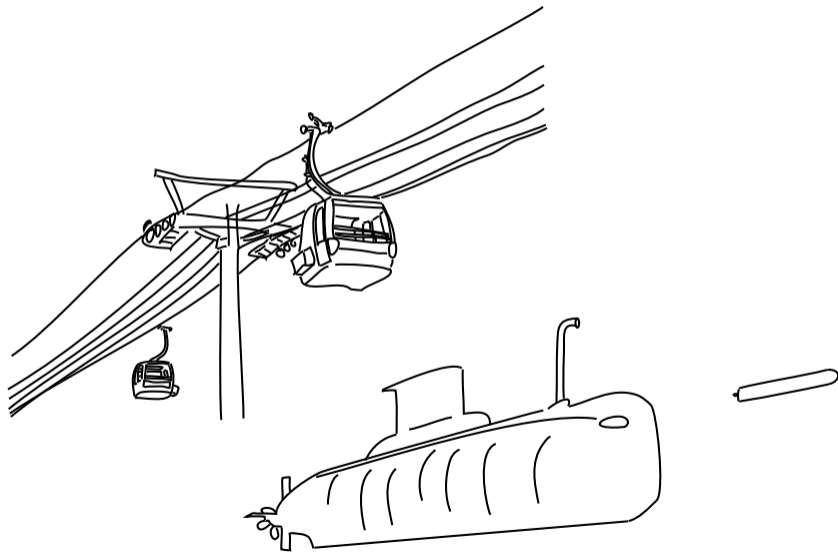
Reflexionemos

# Lista enlazada

- Inserción/extracción rápidas
- Acceso aleatorio... leeento



# ¿Cómo compararlos?



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos



# Sugerencia de Jon Bentley



```
for ( ; ; )  
{
```

Acceso  
aleatorio

Inserción /  
extracción

```
}
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Sugerencia de Jon Bentley

“Insert a sequence of random integers  
into a sorted sequence,

then **remove** those elements **one by one**  
as determined by  
a **random** sequece of **positions**”

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

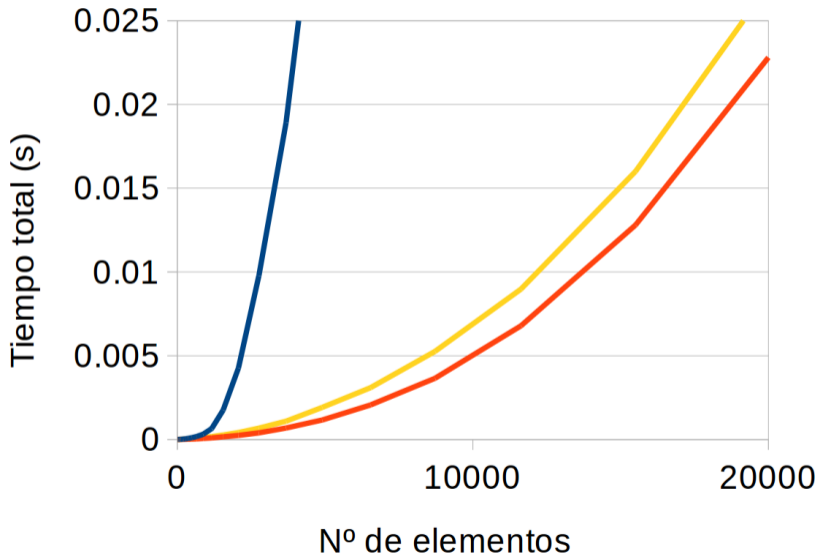
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Resultado



— list  
— vector  
— deque

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Conclusión

El vector es más rápido  
en cierta proporción fija  
(una proporción considerable)

Pero...

¿Nos interesa realmente  
el problema de Jon Bentley?

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Super Árbol

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

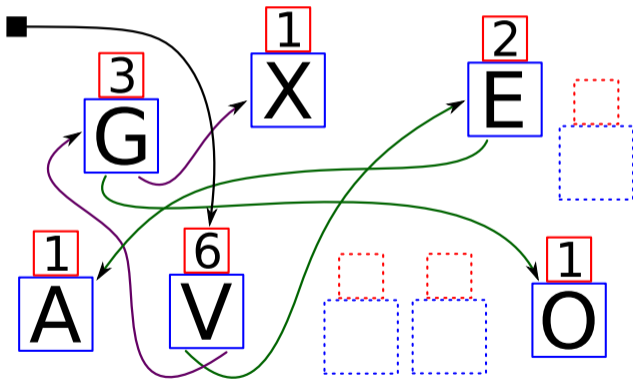
Aplicaciones

Propuestas  
similares

Reflexionemos

# Árbol aumentado (enredado)

Como una lista, pero con dos "siguientes" (izq., y der.)



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

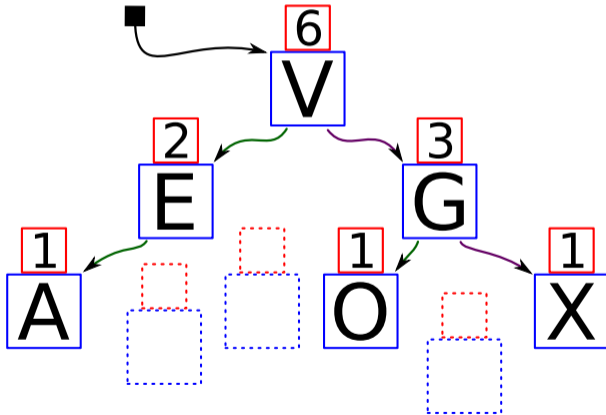
Aplicaciones

Propuestas  
similares

Reflexionemos

# Árbol aumentado

Metadato especial: número de nodos del sub-árbol



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Acceso aleatorio (1/3)

```
template <typename T>
struct node
{
    node<T> * left;        // Sub-árbol izq.
    node<T> * right;      // " " der.
    std::size_t count;    // Núm. nodos
    T value;              // Carga útil
};
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos



# Acceso aleatorio (2/3)

```
template <typename T>
node<T> * RandomAccess (node<T> * root,
                        std::size_t pos)
{
    if (pos >= root->count)
        return nullptr;

    node<T> * p = root;
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Acceso aleatorio (3/3)

```
for (;;)
{
    std::size_t nLeft = p->left ?
                        p->left->count : 0;

    if (pos == nLeft)           return p;
    else if (pos < nLeft)      p = p->left;
    else // (pos > nLeft)
    {
        pos -= nLeft + 1;
        p = p->right;
    }
} // fin
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

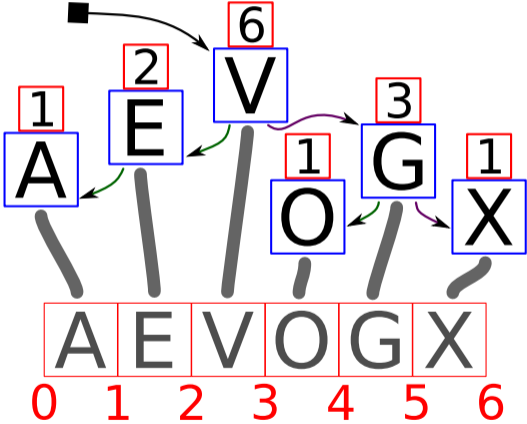
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Vista proporcional



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Complejidad computacional

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

	Acceso aleatorio	Inserción/ Extracción	<b>Suma de ambas</b>
Array	$O(1)$	$O(N)$	$O(N)$
Lista	$O(N)$	$O(1)$	$O(N)$
Súper Árbol	$O(\log(N))$	$O(\log(N))$	$O(\log(N))$

# Complejidad computacional (leyenda)

- $O(1)$  = constante 🤖
- $O(\log(N))$  = logarítmica 😊
- $O(N)$  = lineal 😐
- $O(N \log(N))$  = “linearítmica” 😞
- $O(N^c)$  = polinómica 😡
- $O(c^N)$  = exponencial 🤯
- $O(N!)$  = factorial 🌀

**$N$** : tamaño del problema,     **$c$** : constante  $> 1$

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol










Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Complejidad computacional

	Acceso aleatorio	Inserción/ Extracción	<b>Suma de ambas</b>
Array			
Lista			
Súper Árbol			

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares







Reflexionemos

# Complejidad computacional

---

$$(1 \text{ acc. aleatorio} + 1 \text{ ins./extr.}) \times N = \mathbf{total}$$

---

Array		
Lista		
Súper Árbol		

---

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

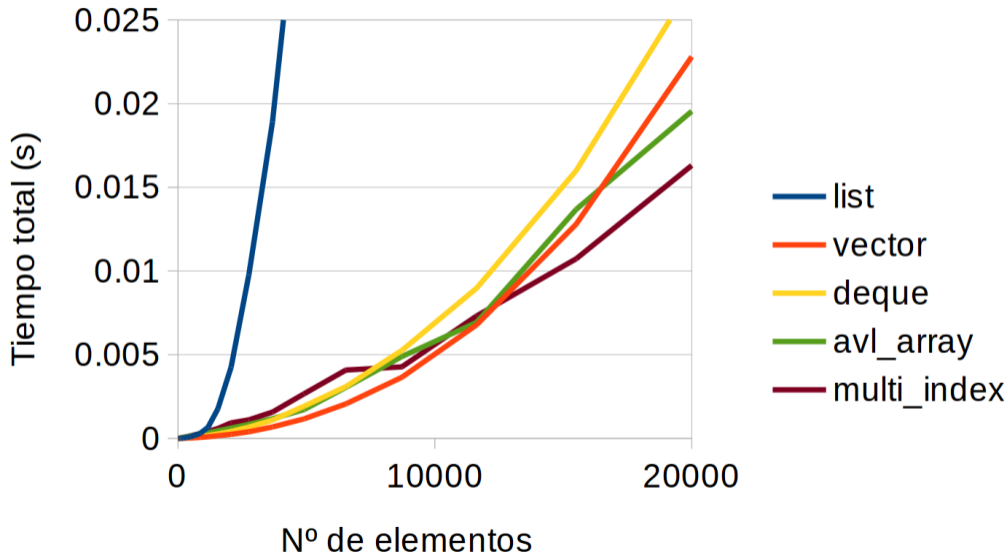
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Resultado (1/3) — pocos elementos



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

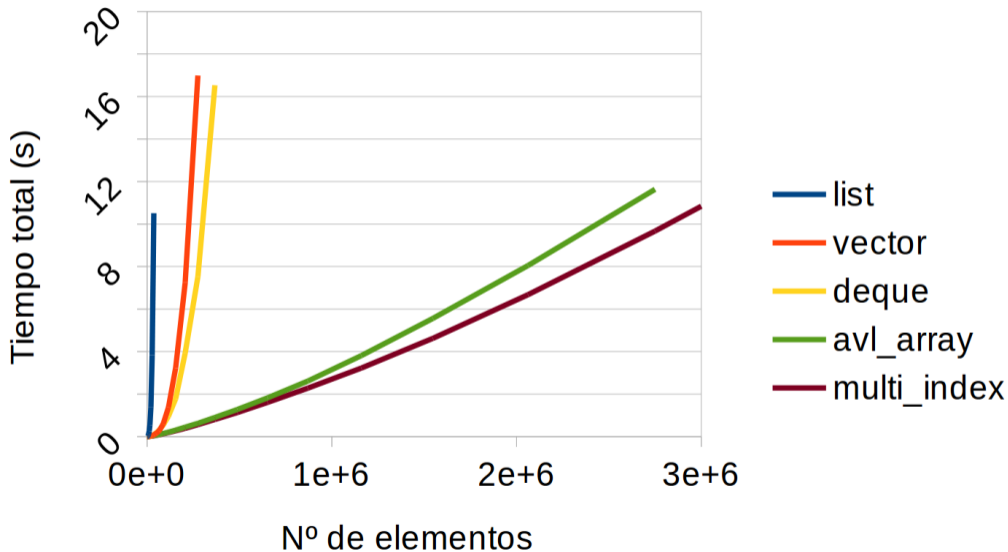
Aplicaciones

Propuestas  
similares

Reflexionemos



# Resultado (2/3) — muchos elementos



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

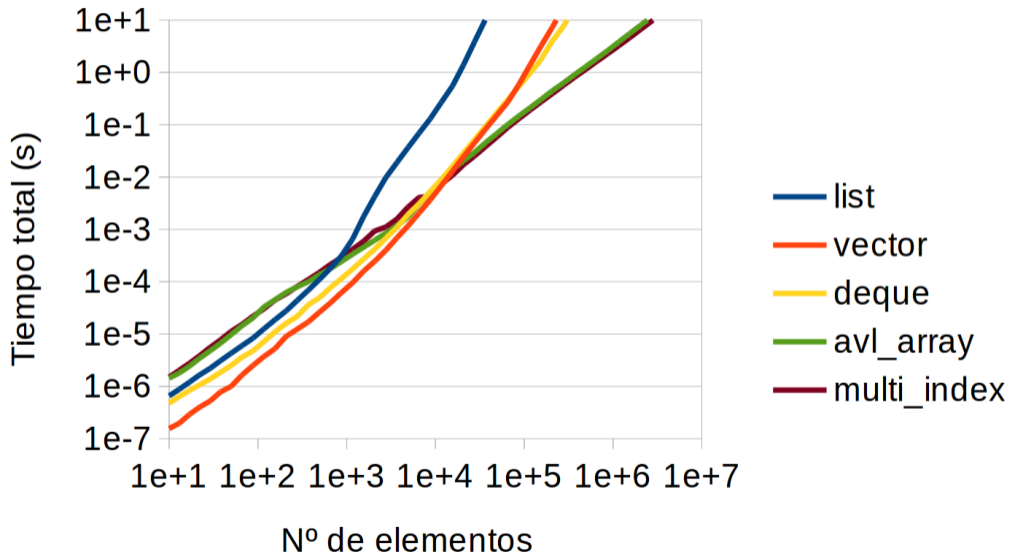
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Resultado (3/3) — escala logarítmica



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

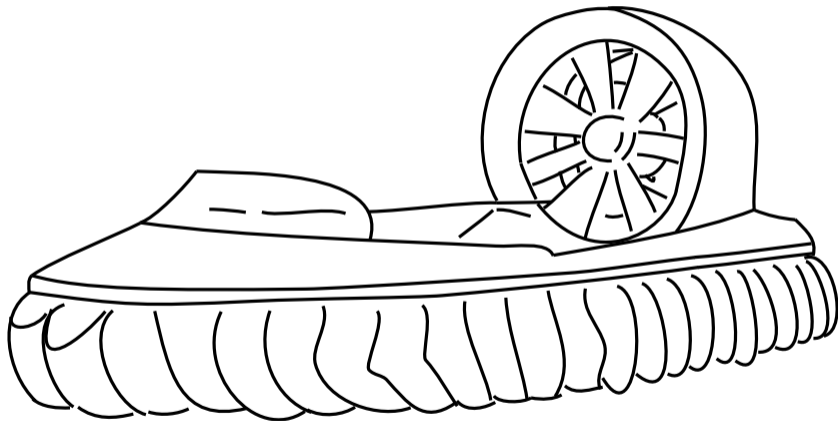
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Ideal para la playa



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Vista no proporcional

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

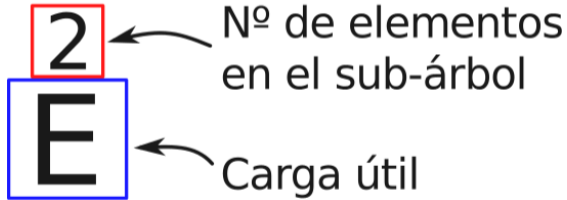
Aplicaciones

Propuestas  
similares

Reflexionemos

# Leyenda en vista proporcional

1 elemento = 1 unidad



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

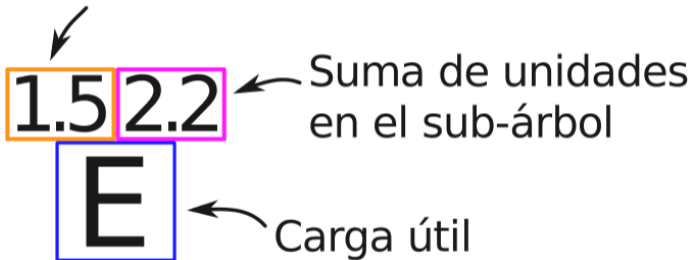
Aplicaciones

Propuestas  
similares

Reflexionemos

# Leyenda en vista **no** proporcional

**este elemento = 1.5 unidades**



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

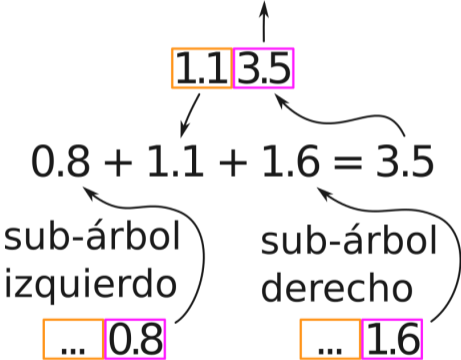
Vista no  
proporcional

Aplicaciones

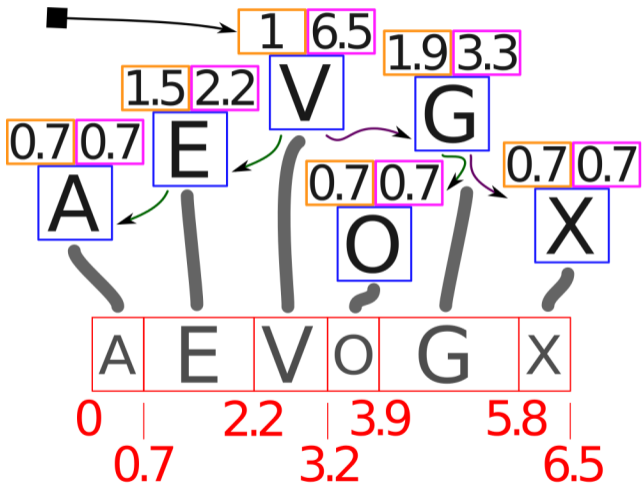
Propuestas  
similares

Reflexionemos

# Suma en vista **no** proporcional



# Vista **no** proporcional



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos



# Aplicaciones

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Editor de texto

## Secuencia de líneas

- Número de bytes
- Número de líneas tras el ajuste
- Número de caracteres
- Si no es texto plano, número de píxels

gtk

Árbol B+ “ad hoc” con número de caracteres y líneas

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

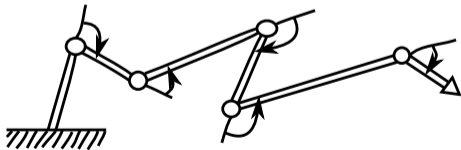
Aplicaciones

Propuestas  
similares

Reflexionemos

# Brazo robot o cadena de moléculas

- Secuencia de transformaciones de traslación y rotación
- Operación de vista **no** proporcional: suma y producto de matrices



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Versión en disco: `shiftable_files`

- Implementación basada en proyección de archivos en memoria
- Código horrible (¡macros!)
- Metadatos en el propio archivo
- Al cerrar se puede optar por:
  - 1 Recompactar el archivo, ó...
  - 2 dejarlo tal cuál, con los metadatos

¿Cómo seguir el rastro de las secciones?

Con una secuencia en memoria con vista **no** proporcional

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Edición de archivos XML gigantes

- Una primera pasada puede construir un índice en memoria (no necesariamente completo)
- Se puede insertar/extraer nodos sin reescribir todo el archivo
- Hay que mantener el índice al día, claro
- ¿Recompactar al cerrar?
  - 1 Sí: vuelve a ser un XML normal
  - 2 No: más rápido

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Propuestas similares

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Multi Index (1/2)

```
boost::multi_index_container  
<  
    T,  
    boost::multi_index::indexed_by  
    <  
        boost::multi_index::ranked_non_unique  
        <  
            boost::multi_index::identity<T>,  
            unordered_less<T>  
        >  
    >  
>  
>
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Multi Index (2/2)

```
template<typename T>
struct unordered_less
{
    bool operator() (const T &,
                    const T &) const
    {
        return false;
    }
};
```

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos



# Propuestas similares en Boost (1/2)

- 2004 – La mención más antigua (no sé si implementada), por Peter Palotas

<http://lists.boost.org/Archives/boost/2004/03/62823.php>

- 2006 – “Hierarchical Data Structures” por Bernhard Reiter y René Rivera

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n2101.html#tr.hierarchy.augment>

- 2006 – “AVL Array” (horrible nombre, lo sé)

<http://sourceforge.net/projects/avl-array>

“Rank List” tras debate en foro de Boost

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Propuestas similares en Boost (2/2)

- 2012 – Countertree por Vadim Stadnik  
[http://dl.dropbox.com/u/8437476/works/  
countertree/doc/index.html](http://dl.dropbox.com/u/8437476/works/countertree/doc/index.html) (enlace roto)
- 2015 – SegmentedTree por Chris Clearwater  
[https://det.github.io/segmented\\_tree/](https://det.github.io/segmented_tree/)

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Propuestas similares fuera de Boost

- “Simon Tatham’s Algorithms Page”

<https://www.chiark.greenend.org.uk/>

[~sgtatham/algorithms/cbtree.html](https://www.chiark.greenend.org.uk/~sgtatham/algorithms/cbtree.html)

“Counted B-trees: An enhancement to the well known B-tree algorithms to allow you to **look up items in the tree by numeric index**, or to **find the numeric index of an item**. Useful for finding percentiles, [...]”

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Propuestas similares en Python

- <https://pypi.python.org/pypi/rbtree>
- <https://pypi.python.org/pypi/pyavl>
- <https://pypi.python.org/pypi/blist>

Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

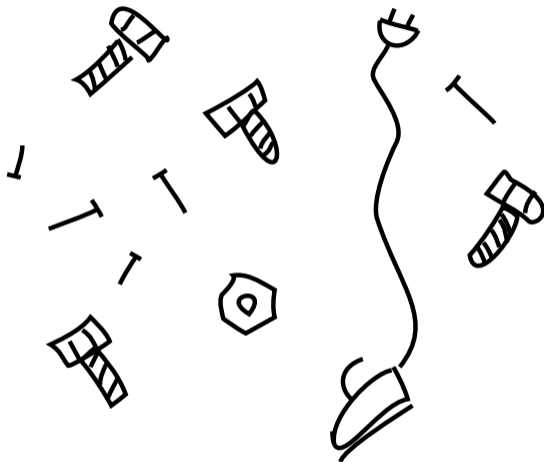
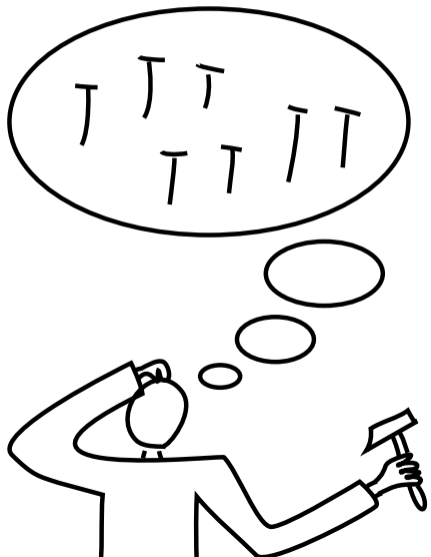
Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Reflexionemos



Super Árbol

Martín K.R.  
indizen

Intro

Super Árbol

Vista no  
proporcional

Aplicaciones

Propuestas  
similares

Reflexionemos

# Muchas gracias

# ¿Alguna pregunta?