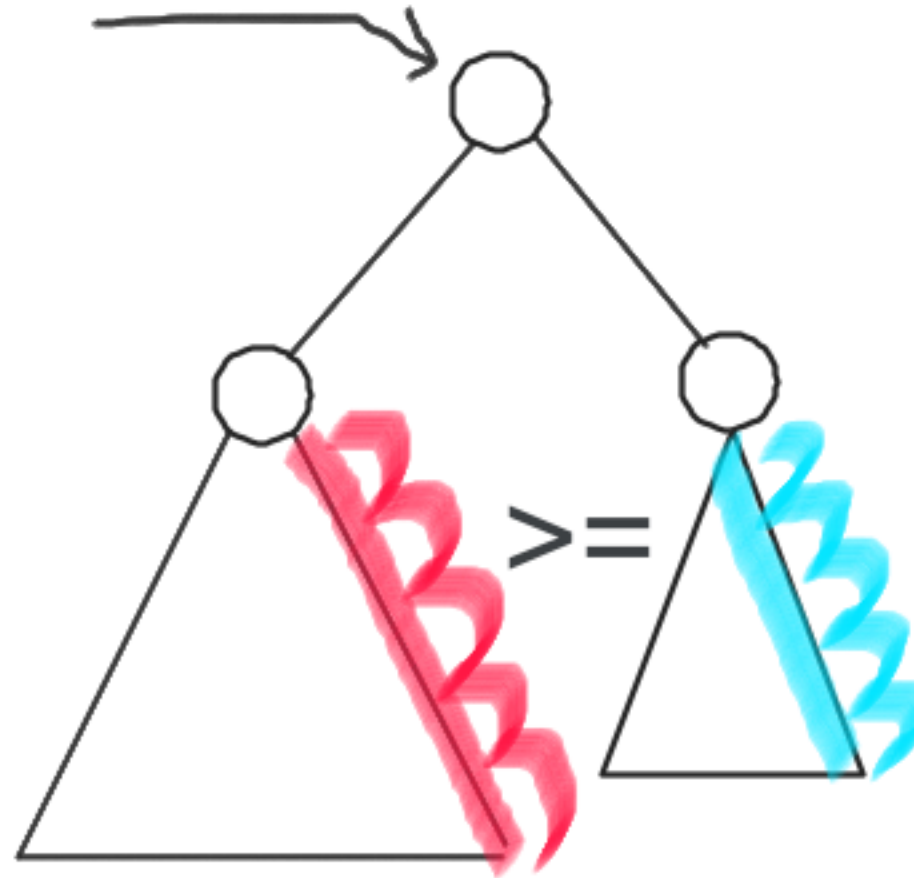


Invariante Leftist: el rango de cualquier hijo izquierdo es mayor o igual que el de su hermano de la derecha.

Rango es la longitud de la espina derecha de cada nodo.



```

type Rank = Int
data Heap a = E | N Rank a (Heap a) (Heap a)

```

Definimos la función *merge*:

```

merge :: Ord a => Heap a -> Heap a -> Heap a
merge h1 E = h1
merge E h2 = h2
merge h1@(N _ x a1 b1) h2@(N _ y a2 b2) =
    if x <= y then makeH x a1 (merge b1 h2)
    else makeH y a2 (merge h1 b2)

```

```

rank :: Heap a -> Rank
rank E = 0
rank (N r _ _) = r

```

Definimos *makeH*:

```

makeH x a b = if rank a >= rank b then N (rank b + 1) x a b
              else N (rank a + 1) x b a

```

Luego tenemos que:

- La espina derecha es la ruta más corta a una hoja.
- Los elementos de la espina derecha están ordenados.

