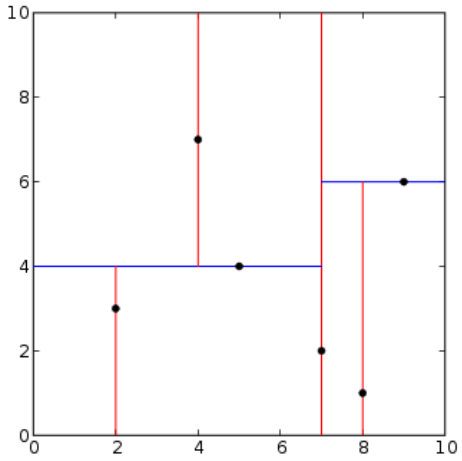
Trabajo Práctico 1

Un espacio métrico es un conjunto M (cuyos elementos se denominan puntos), junto con una función distancia $d : M \times M \to \mathbb{R}$ (llamada métrica) que satisface ciertas propiedades atribuídas a la distancia. Algunos ejemplos de espacios métricos son los números reales con la función distancia $d(x, y) = |y - x|$, o más generalmente el espacio euclídeo de n dimensiones, el cual está formado por el espacio vectorial sobre las tuplas de n elementos $(x_1, \ldots, x_n)$ en donde cada $x_i \in \mathbb{R}$ junto con la función distancia entre dos puntos $(x_1, \ldots, x_n)$ e $(y_1, \ldots, y_n)$ definida como: $\sqrt{\sum_{i=0}^{n}(y_i - x_i)^2}$.

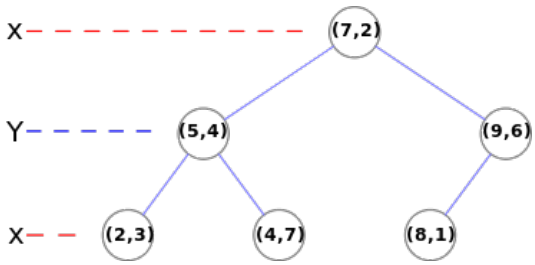
Para almacenar un conjunto de puntos de un espacio métrico de n dimensiones, suele ser útil una estructura de datos que organiza los puntos del espacio en un árbol binario, el cuál puede pensarse como una generalización de los árboles binarios de búsqueda.

Este tipo particular de árboles emplea planos perpendiculares a uno de los ejes del sistema de coordenadas, de la siguiente manera, cada nodo del árbol almacena un punto p de dimensión n y un eje e, de manera que el hiperplano que pasa por p y es perpendicular al eje e divide al espacio en dos semiespacios y los puntos que se encuentran a un lado de este hiperplano (con la componente correspodiente al eje de coordenada e menor o igual al de p) son representados por el subárbol izquierdo, y los que se encuentran del otro lado del hiperplano por el subárbol derecho.

En los siguientes gráficos se muestra un ejemplo de cómo se pueden almacenar en un árbol de estas características, los puntos del plano: (2,3), (5,4), (9,6), (4,7), (8,1) y (7,2).



El punto (7,2) divide al plano usando el eje x, con $x = 7$, quedando en un semiplano los puntos (2,3), (5,4) y (4,7) y en el otro los puntos (9,6) y (8,1). El punto (5,4) divide el plano con $y = 4$, quedando el punto (2,3) en un semiplano y (4,7) en el otro. Por otro lado, el punto (9,6) divide el plano usando el eje y, quedando el punto (8,1) en un semiplano y ninguno en el otro.
El árbol resultante es el siguiente:



Para representar esta clase de árboles en Haskell se utilizará el siguiente tipo de datos:

```
data NdTree p = Node (NdTree p)   -- subárbol izquierdo
                     p            -- punto
                     (NdTree p)   -- subárbol derecho
                     Int          -- eje
              | Empty
  deriving (Eq, Ord, Show)
```

1. Para representar puntos de un espacio métrico n-dimensional, se utilizará la siguiente clase:

```
class Punto p where
  dimension :: p → Int          -- devuelve el número de coordenadas de un punto
  coord :: Int → p → Double     -- devuelve la coordenada k-ésima de un punto (comenzando de 0)
  dist :: p → p → Double        -- calcula la distancia entre dos puntos
```

   a) Definir la función *dist* :: *Punto* $p \Rightarrow p \to p \to$ *Double* que calcula la distancia entre dos puntos $p = (x_0, \ldots, x_n)$ y $q = (y_0, \ldots, y_n)$ usando la siguiente fórmula:

$$\sum_{i=0}^{n} (y_i - x_i)^2$$

   b) Se utilizarán los siguientes tipos de datos para representar puntos en el plano y en el espacio tridimensional:

```
newtype Punto2d = P2d (Double, Double)
newtype Punto3d = P3d (Double, Double, Double)
```

   Dar las instancias de *Punto* para *Punto2d* y *Punto3d*.

2. Para generar un árbol de puntos a partir de una lista de puntos de un espacio n-dimensional, se utilizará el siguiente método:

   I- Seleccionar el eje sobre el cual se alineará el hiperplano. Para seleccionar un eje distinto en cada paso, elegir el eje de coordenadas $level \,\%n$ (donde % es el operador módulo), donde *level* es el nivel del árbol que se está construyendo.

   II- Calcular la mediana de la lista de puntos, según el eje seleccionado.

   III- Utilizar los puntos de la lista menores o iguales a la mediana (según el eje seleccionado), para crear un árbol l, aplicando este método.

   IV- Hacer lo mismo para crear un árbol r, con los puntos mayores a la mediana.

   V- Crear un nodo del árbol que tenga como raíz la mediana, como eje el eje seleccionado, como subárbol izquierdo el árbol l y como subárbol derecho el árbol r.

   Definir una función *fromList* :: *Punto* $p \Rightarrow [p] \to$ *NdTree* p, que construya un árbol a partir de una lista de puntos, utilizando el método dado.

3. Se pueden añadir nuevos elementos a este tipo particular de árboles de manera similar a la que se añaden a los árboles binarios de búsqueda. Se recorre el árbol empezando por la raíz y siguiendo por el nodo de la izquierda o de la derecha dependiendo de si el punto que se quiere añadir está de un lado o del otro del hiperplano de corte. Una vez que se llega a un nodo hoja, se añade el nuevo punto a la izquierda o a la derecha de éste, de nuevo dependiendo del hiperplano de corte.

   Definir una función *insertar* :: *Punto* $p \Rightarrow p \to$ *NdTree* $p \to$ *NdTree* p, que agregue un nuevo punto a un árbol.

4. Para eliminar un elemento de un árbol de tipo *NdTree* existen varios enfoques. Uno de ellos consiste en encontrar un elemento que reemplace al nodo a eliminar. Sea r el punto que se desea eliminar del árbol, primero se busca r en el árbol, si r es una hoja, se termina sin reemplazar el nodo eliminado. Sino, se busca un reemplazo para r de la siguiente manera, si r está asociado a un eje e y tiene subárbol derecho, se busca en éste el punto con menor valor en la componente correspodiente al eje e, sino se busca el reemplazo en el subárbol izquierdo con mayor valor en la componente correspodiente a e.

   Definir una función $eliminar :: (Eq\ p, Punto\ p) \Rightarrow p \rightarrow NdTree\ p \rightarrow NdTree\ p$, que elimine un punto a un árbol.

5. Una de las ventajas de organizar los puntos de un espacio utilizando este tipo particular de árboles, es que se pueden realizar búsquedas de puntos próximos a un punto dado, o búsquedas de puntos que se encuentran en una región dada, con bajos costos.

   Suponiendo un espacio de 2 dimensiones, el siguiente tipo de datos puede utilizarse para representar un rectángulo en el mismo:

   type $Rect = (Punto2d, Punto2d)$

   Definir una función $ortogonalSearch :: NdTree\ Punto2d \rightarrow Rect \rightarrow [Punto2d]$, que dado un conjunto s de puntos en el plano y un rectángulo, encuentre los puntos de s que están dentro del rectángulo dado. Definir la función de manera eficiente, utilizando la información sobre los ejes de los nodos.