Clase de Laboratorio:

24 de Mayo

Vamos a implementar conjuntos mediante de datos eficientes. Comlejidad de las estructuras implementadas

hasta ahora:

	arregki red	lista enlazada	lista bi enlazada
pertenece	$O(logN)$	$O(N)$	$O(N)$
insertar	$O(N)$	$O(N)$	$O(N)$
borrar	$O(N)$	$O(N)$	$O(N)$
max/min	$O(1)$	$O(N)/O(1)$	$O(N)/O(1)$

Arboles binarios: Cada nodo un solo padre y hasta dos hijos, hijos a la izquierda menores y a la derecha mayores.
Invariante de representacion de un arbol binario:
pred InvRep (arbol){
un objeto es ABB $\leftrightarrow$ es null $\vee$ todos los nodos a la derecha son mayores y los de la izquierda son menores, y estos tambien son arboles de busqueda
}
El objetivo sera implementar un conjunto mediante arboles binarios de busqueda.

Algoritmos
abb.pertenece

```
if raiz.valor==elemento then
   return true
else
   if elemento<raiz.valor then
      raiz.izquierda.pertenece(elemento)
   else
      raiz.derecha.pertenece(elemento)
   end if
end if
```

abb.insertar(elem) Parecido al anterior
(porque devuelve el ultimo nodo de la busqueda)
Si lo encontramos no hacemos nada
Si NO lo encontramos lo insertamos como hijo del ultimo nodo buscado

eliminar
recorrer y si esta eliminarlo si no no hacerlo,
si esta y tenemos que eliminarlo, si el nodo no tiene hijos lo eliminamos directamente.
si tiene un unico elemento, enlazar el padre del nodo que queremos eliminar al hijo de este mismo.
si tiene dos hijos, buscamos el mayor de los hijos menores, o el menor de los hijos mayores, y lo eliminamos y lo reemplazamos en lugar del padre