Facultad de Ciencias Exactas, Ingenier Agrimensura
Instituto Politecnico Superior
Analista Universitario en Sistemas
Estructuras de Datos y Algoritmos II

Nombre y Apellido: Legajo:

Examen

1. Un vector es una estructura de datos indexada, que tiene una dimensión fija y permite agregar elementos en posiciones aleatorias dentro del rango de indices. Se considera que los indices comienzan en 0. Las siguientes operaciones son soportadas por este TAD:

```
tad Vector (A : Set) where
  import Bool, Int, Maybe
  ini     : Int → Vector A
  ins     : Int → A → Vector A → Vector A
  view    : Int → Vector A → Maybe A
  size    : Vector A → Int
  reverse : Vector A → Vector A
```

- *ini* dado un entero retorna un vector de esa dimensión sin elementos (vacío)
- *ins* inserta un elemento en la posición dada, si la posición no es válida no se inserta
- *size* muestra la dimensión del Vector
- *view* muestra un elemento en una posición dada, si la posición no es válida o no contiene elemento retorna Nothing
- *reverse* dado un vector retorna el vector en orden inverso.

Dar las especificación algebraica del TAD Vector.

2. Usando los siguiente tipo:

```
type Tam = Int
data Vector a = E | N Tam (Vector a) (Maybe a) (Vector a)
```

Implementar en haskell las funciones *ini*, *ins* y *view* del TAD Vector del ejercicio anterior. Donde:

- *ini* n deberá construir un árbol balanceado vacío de dimensión n. Usaremos para representar un indice vacío el tipo Maybe a.
  Ejemplo *ini* 3 = N 3 (N 1 E Nothing E) Nothing (N 1 E Nothing E)
- *insert* n e insertará un elemento e en la posición n del árbol, respetando el recorrido *inorder* del árbol.
  Ejemplo: *ins* 2 `'c'$ins 0 'a'$ins 1 'b'` (*ini* 3) = N 3 (N 1 E (Just `'a'`) E) (Just `'b'`) (N 1 E (Just `'c'`) E)
- para la implementación de las funciones no se convertirá el tipo Vector en una lista, ni una lista en el tipo Vector.

3. Dadas las siguientes definiciones:

```
elem x []       = False          or []       = False
elem x (y : ys) = x ≡ y ∨ elem x ys   or (x : xs) = x ∨ or xs
```

a) Enunciar el principio de inducción estructural para $[a]$

b) Probar por inducción estructural sobre $elem\ x\ (concat\ xss) = \mathsf{or}\ (map\ (elem\ x)\ xss)$

4. Dada la siguiente recurrencia:

$$W_f(1) = 1$$
$$W_f(n) = 3W_f(\frac{n}{2}) + n^2$$

a) Resolver la recurrencia expandiendo la definición en forma algebraica.

b) Utilizando lo resuelto en el ejercicio anterior dibujar el árbol de recurrencia con los valores por niveles y realizar las operaciones necesarias para alcanzar los mismos valores.

c) De ser posible confirma el resultado del primer ítem utilizando el teorema maestro o por el contrario justificar por que no es posible usarlo.