

RECUPERATORIO PARCIAL 2

Cátedra Estructuras de Datos I

22 de Junio de 2017

1. Presentación del ejercicio

En el día de ayer se comunicó la ministra de Seguridad de la Nación, la Dra. Patricia Bullrich para organizar la asistencia al evento del día 20 de junio en la ciudad de Rosario. La idea es poder tener un control y un orden que permita saber los invitados que asistirán al evento nos comentó. De cada asistente se va a almacenar su dni, nombre y apellido, la descripción del cargo y una prioridad que debe ser usada para, de alguna manera, tener un cierto orden entre los invitados pero, ¡no queremos una lista enlazada!



```
typedef struct {
    char* dni, *nombre, *descripcionCargo;
    int prioridad;
} _Invitado;

typedef _Invitado* Invitado;
```


Lista de árboles	
1. Algarrobo	(Pag. 1)
2. Azúcar	(Pag. 15)
3. Calamita	(Pag. 22)
4. Cacaotero	(Pag. 28)
5. Cedro	(Pag. 38)
6. Ceibo	(Pag. 50)
7. Eucalipto	(Pag. 60)
8. Flacopando	(Pag. 70)
9. Ficus	(Pag. 80)
10. Geraniella	(Pag. 90)
11. Higuera	(Pag. 100)
12. Huanahuanay	(Pag. 110)
13. Jabonera	(Pag. 120)
14. Jacaranda	(Pag. 130)
15. Liriodendro	(Pag. 140)
16. Maité	(Pag. 150)
17. Mimosa	(Pag. 160)
18. Mopora	(Pag. 170)
19. Mulo de monte	(Pag. 180)

Quiero que se deje de pensar en una lista secuencial, para pensar en árboles pero, que esto quede claro, no quiero una Lista de Árboles sino, un Arbol! Al no entender nos dijo que no quería que nos confundieramos con esto. Lo que se nos ocurrió fue proponer el uso de un árbol binario para su implementación; a lo cual nos comentó que le parecía excelente.

```
typedef struct _Nodo {
    Invitado invitado;
    struct _Nodo* der, izq;
} Nodo;

typedef Nodo* Arbol;
```

Pero, para complicar el acceso a los datos, se debía poder pasar la información de una estructura de datos a otra. No hay problema, le dijimos, podemos usar un Heap de manera tal que con el mismo construyamos el árbol binario y, a partir del árbol binario si suponemos que es completo, construyamos el heap. Está muy bien realizar una suposición. Podemos asumir que el árbol binario es completo.




```
#define MAX_HEAP 20

typedef struct {
    Invitado datos[MAX_HEAP];
    int nelems;
} _BHeap;

typedef _BHeap* BHeap ;
```

Para poder realizar esto pensamos en los siguientes prototipos para las funciones que se encargarían de realizar estas conversiones:



```
Arbol convierteHeapEnArbol(Bheap bHeap);

BHeap convierteArbolEnHeap(Arbol inicio);
```

y, toda esta información la vamos a almacenar en un archivo con extensión .h.
Luego, nos agradeció por resolver esto a la brevedad y, cortó la comunicación.

2. Evaluación

Se pide que escriba un código que pueda implementar las funciones solicitadas respetando los tipos definidos y los prototipos indicados anteriormente.

Se tomará como criterio de aprobación:

1. funcionamiento del programa;
2. claridad y eficiencia del código escrito;