

Práctica 4: Montículos

Fecha límite de entrega: miércoles, 7 de diciembre

1. Implemente las siguientes operaciones sobre **montículos de máximos**.

```
#define TAM 512000
typedef struct {
    int vector[TAM];
    int ultimo;
} monticulo;

void crear_monticulo(int [], int, monticulo *);
int eliminar_mayor(monticulo *);
```

Valide que las operaciones `crear_monticulo` y `eliminar_mayor` funcionen correctamente.

Nota: Al comenzar los vectores en C a indexarse por cero, las posiciones relativas de los hijos y del padre de un nodo en el vector deben recalcularse con respecto a lo visto en clase de teoría.

2. Demuestre empíricamente que crear un montículo a partir de un vector cualquiera con n elementos (operación `crear_monticulo`) se ejecuta en un tiempo $O(n)$. Para ello se recomienda el uso de secuencias ascendentes de números.
3. Implemente la *ordenación por montículos*: `void ord_monticulo(int [], int)`

```
procedimiento Ordenación por montículos (var V[1..n])
    crear_monticulo (V, M);
    para i := n hasta 1 paso -1 hacer
        V[i] := eliminar_mayor(M);
    fin para
fin procedimiento
```

Valide el correcto funcionamiento del algoritmo de ordenación.

4. Calcule empíricamente la complejidad del algoritmo de ordenación para tres situaciones iniciales diferentes: (a) el vector ya está ordenado en orden ascendente, (b) el vector ya está ordenado en orden descendente, y (c) el vector está inicialmente desordenado.
5. Entregue los ficheros con el código C y el fichero `.txt` con el informe por medio de la tarea *Entrega Práctica 4* en la página de Algoritmos en <https://campusvirtual.udc.gal>. Se recuerda que el límite para completar la tarea es el miércoles 7 de diciembre a las 23:59, y una vez subidos los archivos no se podrán cambiar. **Todos los compañeros que forman un equipo tienen que entregar el trabajo.**