

**Práctica 3: Montículos**

Fecha límite de entrega: sábado, 11 de noviembre

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

```
#define TAM 256000

struct monticulo {
    int ultimo;
    int vector[TAM];
};

typedef struct monticulo * pmonticulo;

void crearMonticulo(int v [], int n, pmonticulo m);
int quitarMenor(pmonticulo m);
```

Valide que las operaciones anteriores 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 `crearMonticulo`) se ejecuta en un tiempo  $O(n)$ .
3. Implemente la *ordenación por montículos*: `void ordenarPorMonticulos(int v [], int n)`

```
procedimiento OrdenarPorMontículos (var V[1..n])
    crearMonticulo (V, M);
    para i := 1 hasta n hacer
        V[i] := quitarMenor(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 3* en la página de Algoritmos en <https://campusvirtual.udc.gal>. Se recuerda que el límite para completar la tarea es el sábado 11 de noviembre 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.**