

Práctica 3: Montículos

Fecha límite de entrega: jueves, 21 de noviembre

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

```
#define TAM 256000
typedef struct {
    int vector[TAM];
    int ultimo;
} monticulo;

void crearMonticulo(int [] v, int n, monticulo *m);
int consultarMenor(const monticulo *m);
void quitarMenor(monticulo *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 ordMonticulo(int [], int)`

```
procedimiento Ordenación por montículos (V[1..n])
    crearMonticulo (V, M) ;
    para i := 1 hasta n hacer
        V[i] := consultarMenor(M) ;
        quitarMenor(M)
    fin para
fin procedimiento
```

Compruebe que el algoritmo de ordenación funcione correctamente.

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 informe (archivo txt) en una carpeta **P3** mediante SVN en el repositorio <https://svn.fic.udc.es/grao2/alg/19-20/pepe.perez> (sustituya *pepe.perez* por su login).